Epidemiologic Considerations Related to the Use of the Prognostic Score in Selection Bias and Confounding.

by

Keri L. Calkins

A thesis submitted to Johns Hopkins University in conformity

with the requirements for the degree of

Master of Science

Baltimore, Maryland

April 2014

Abstract

Background: The prognostic score is a method similar to the propensity score that estimates the conditional probability of the outcome. The existing methods to correct for selection bias are limited when dealing with non-participation in forming the study population. An application to selection bias has not been explored.

Objective: This thesis explores the use of the prognostic score as a method to address confounding and selection bias. Multiple forms of the prognostic score will be employed to estimate the unbiased effect in the presence of confounding. We will also compare the use of the prognostic score in the presence of selection bias to inverse probability of selection weights (IPSW) and direct adjustment.

Design: Based on several directed acyclic graphs, Monte Carlo simulations compared several approaches to isolating the effect estimate. In the presence of confounding, weighting using three variations of the prognostic score were compared to weighting using the propensity score. Approaches to combining the prognostic and propensity score were also investigated. In the presence of selection bias, weighting using the three prognostic score approaches, IPSW, and direct adjustment were compared.

Main Outcomes: Percent relative bias, robust variance estimates, Monte Carlo variance estimates, and MSE with respect to the marginal and conditional odds.

Results: In the presence of confounding, the stabilized modified prognostic score weights and stabilized IPEW yielded the marginal odds ratio, while the combination prognostic and propensity score approaches, the unexposed

prognostic score weights, and the full population prognostic score weights resulted in the conditional odds ratio. For the selection bias simulations, the unexposed and full population prognostic score weights estimated the conditional odds ratio and were comparable to direct adjustment methods. The modified prognostic score yielded a result that appeared to be a mix of the marginal and conditional odds ratio. In the presence of unmeasured selection variables, the prognostic score approaches and direct adjustment were biased.

Conclusions and Relevance: The prognostic score is an acceptable alternative method to adjust for confounding and for selection bias except for when the selection variable acts as a collider in the presence of unmeasured variables.

Thesis Readers:

Bryan Lau, PhD MS MHS

Elizabeth Stuart, PhD AM

Acknowledgements

I would first like to thank my advisor and thesis reader, Dr. Bryan Lau, for his mentorship and support with this project and our other collaborations. Bryan was instrumental in my conceptualization and execution of this thesis. I look forward to our continued work together throughout my studies in the doctoral program. I would also like to thank my thesis reader, Dr. Elizabeth Stuart, for her insights and thorough review of this project. Her prior work with the propensity score and the prognostic score greatly enhanced my understanding and provided a foundation for the methods explored in this thesis.

I am grateful for the experiences I have had at Johns Hopkins as both an undergraduate and graduate student. I am particularly grateful for my experiences as a student at the Bloomberg School of Public Health. The students and faculty at Bloomberg are among the most dedicated, collegial, and innovative individuals I have ever met. Finally, I would like to thank my friends and family for their continued love and support.

Table of Contents

Chapter 1

Chapter 2

Chapter 3

Chapter 1: Introduction

1.1 Background

The background section addresses the history of the prognostic score and the issue of selection bias in epidemiologic studies in two separate sections. Ultimately, the thesis will address two applications of the prognostic score in epidemiologic research: the isolation of the effect estimate in the presence of confounding and the isolation of the effect estimate in the presence of selection bias.

Prognostic Score

The prognostic score is an epidemiologic tool originally defined as the multivariate confounder score by Miettinen in 1976 as an alternative to the use of stratification and multivariate models to address confounding.[1] In an effort to capitalize on the strengths of stratification and multivariate models, Miettinen proposed stratification based on the "multivariate confounder score" thereby accounting for a number of confounders through stratification on a single metric.[1] It was proposed that the multivariate confounder score be derived from either a scoring function based on the outcome conditional on being unexposed or a scoring function based on the exposure conditional on being a non-case.[1] Miettinen indicates a preferences for using an outcome-based scoring function and uses this procedure in the application section of the paper where the score is derived by fitting a model using the "noniterative least squares procedure" of the outcome based on a variety of presumed confounders and the exposure of

interest and predicting the multivariate confounder based on the fitted model while fixing the exposure variable to unexposed.[1]

Hansen's publication revisited and expanded upon the multivariate confounder score in 2008 using updated terminology, the prognostic score.[2] Hansen explores the theoretical application of the prognostic score to confounding in a comparison to the more widely used propensity score and their applications to randomized clinical studies.[2] Hansen specifies that a score "is a prognostic score if and only if conditioning on it induces prognostic balance within the domains determined by X", where X denotes the vector(s) of covariates.[2] Prognostic balance is defined as "similarity among the covariate distributions of trials for subjects with contrasting potential outcomes."[2] The paper also determines the potential need to estimate the prognostic score among the control group rather than the full study population in order to avoid a potential estimation of mixture of the propensity and prognostic scores if the treatment increases the outcome.[2] Ultimately, the paper highlights the potential of the prognostic score as an additional or companion method to address confounding along with the propensity score.[2]

Most recently, Arbogast and Ray conducted a simulation study comparing different methods for addressing bias from multiple confounders in their 2011 publication.[3] The publication compared effect estimate using the propensity score, multivariate regression, and effect estimation using a method defined as the "disease risk score".[3] Arbogast and Ray defines the disease risk score as an estimate of "the probability or rate of disease occurrence as a function of the

covariates" and states that it can be estimated using the method described by Miettinen, labeled the "full-cohort disease risk score", or using the method described by Hansen, labeled as the prognostic score or "unexposed-only disease risk score", where the score is estimated only in the unexposed group.[3] Using Monte Carlo simulations, effect estimation using disease risk scores were found to be comparable to effect estimation using the propensity score and multivariate adjustment.[3] Interestingly, the full-cohort disease risk score seemed to perform better than the unexposed-only disease risk score given that the additional assumption of that the covariates were not effect modifiers was met.[3] Regardless of the estimation approach, Arbogast and Ray highlighted the utility of the disease risk score in situations where the propensity score does not perform well including "exposures that are rare or that have a large number of categories".[3]

The papers by Miettinen, Hansen, and Arbogast detail the various approaches to defining and utilizing a balancing score that for the purposes of this thesis will be called the prognostic score. The choice of this terminology is based on the distinction between the disease risk score and the prognostic score that is detailed by Hansen[2] and by a subsequent Arbogast publication.[4] Stuart, Lee, and Leacy's 2013 paper delineates the distinction between the terminology, describing how the prognostic score "generalizes and extends the unexposed-only disease risk score to continuous, categorical, and ordinal outcomes" as explored by Arbogast,[4] while the unexposed-only disease risk score is a special case of the prognostic score where the outcome is binary.[5] Table 1 provides a

summary of the definitions, equations, and history of the prognostic score and its precursors. Note that the author and year introduced corresponds to the formula and description of the various scores while the original terminology introduced by that author may not be included. Table 1 also provides a column to indicate the type of variable that the prognostic score variants can estimate in order to clarify the distinctions in terminology.

Perhaps as evidenced by the variation in terminology, the prognostic score and its derivations have not been widely used in epidemiologic research. A recent systematic review by Tadrous et al. found 97 unique publications between 1976 and 2010 focused on the disease risk score (DRS).[6] Of these 97 publications, 86 were isolated as applications of the disease risk score to confounding while the remaining 11 were methodological reviews.[6] It is of note that this review did not include the term prognostic score in its search criteria[6], though Hansen introduced this terminology in 2008.[2] The 86 applications in the systematic review were used largely in either observational cohorts (47%) or case-control populations (42%).[6] In 47% of the applications, the DRS were derived using logistic regression, and the overwhelming majority of publications, 93%, used the DRS as a categorical variable.[6] Among the applications that used a categorical DRS, 60% employed stratification and 35% included the categorical DRS in the regression model.[6] Tadrous notes that perhaps these trends were influenced by Pike et al.'s 1979 paper which warned of the potential for overestimation of the effect of the confounders when using logistic regression in the generation of the DRS[7] and the subsequent paper by Cook and Goldman in

1989 which clarifies of the relative rarity of this overestimation when the DRS is used as categorical covariate.[8,6]

Stuart et al.'s simulation study highlighted the role of the prognostic score as a measure of covariate balance and its correlation with bias reduction in the effect estimate.[5] As propensity scores are often calculated without regard to the outcome, prognostic balance can identify potential bias introduced when one or more covariates are strongly associated with the outcome.[5] The simulation compared the correlation between the absolute standardized mean difference (SMD) of the prognostic score and bias in the treatment effect estimate and more traditional balance metrics: absolute SMD of the propensity score, the average absolute standardized mean difference (ASMD), and the Kolmogorov-Smirnov test statistic (K-S Stat).[5] In situations where the covariates that are highly predictive of exposure are also highly predictive of the outcome, all of the measures performed similarly.[5] When the covariates that are highly predictive of the outcome are not the same as those that are predictive of the exposure, the absolute SMD of the prognostic score was superior to the more traditional measures.[5] Stuart et al. concluded that the balance measure based on the prognostic score could be employed to reduce bias in the effect estimate (i.e., could be used to help select an appropriate propensity score approach) and was robust to model misspecification.[5]

Given the performance of the prognostic score when covariates strongly predict the outcome, Leacy and Stuart conducted a further simulation study to correct for bias using the prognostic score in combination with the propensity

score.[9] The simulation explored several methods of combining the prognostic and propensity scores in order to estimate the average treatment effect on the treated (ATT).[9] ATT is defined here as "the difference in potential outcomes amongst those receiving treatment".[9] Several matching and subclassification approaches were used to explore the combination of the prognostic and propensity scores; however, weighting was not examined in the simulation.[9] The combination of the two scores performed well in the simulation and the two full matching approaches yielded a superior estimation of the ATT when compared to matching and subclassification methods based on the individual scores.[9]

Taking into account all of the prior research, the prognostic score and its variants have been applied in experimental and nonexperimental settings, primarily as a means to control for multiple confounders. Recent empirical studies have shown the potential role of the prognostic score in conjunction with the propensity score to address bias in effect measure estimation. While matching, subclassification, and weighting using the propensity score are established approaches,[10] few publications have addressed these methods using the prognostic score.

Table 1: Summary of the Propensity Score, the Prognostic Score and Precursors

	Author and Year Introduced	Description	Formula*	Outcomes
Propensity Score	Rosenbaum, Rubin (1983)[16]	A balancing score where the probability of the exposure/treatment is estimated as a function of the covariates.	Pr(E=1|X=x)	-
Prognostic Score	Hansen (2008)[2]	A balancing score where the probability of the outcome/disease is estimated as a function of the covariates among the unexposed group.	Pr(Y=y|X=x,E=0)	Binary, Continuous, Categorical, and Ordinal Outcomes[5]
Unexposed Disease Risk Score	Miettinen (1976)[1]	A special case of prognostic score where outcome is binary.[2,3]	Pr(Y=1|X=x,E=0)	Binary Outcome
Full Cohort Disease Risk Score (Multivariate Confounder Score)	Miettinen (1976)[1]	A variation of the prognostic score where the probability of having the outcome is regressed on the covariates including the exposure. The score is predicted using the coefficients without respect to the exposure (i.e. E=0).	Pr**(Y=1|X=x,E=e)	Binary Outcome
Modified Prognostic Score	-	A variation of the prognostic score where the probability of having the outcome is regressed on the covariates excluding the exposure. The score is predicted using the coefficients without respect to the exposure (i.e. E=0). This score is equivalent to estimating the propensity score if the outcome were to be assigned as the exposure in the model.	Pr(Y=1|X=x)	-

* E refers to the exposure or treatment, X refers to a vector of covariates, and Y refers to the outcome of interest

** Let Pr**[•]= Pred{E[Y=1|•, E=0]}. (Exposure variable is set to unexposed for prediction of Full Cohort Disease Risk Score).

Selection Bias

Hernán, Hernández-Díaz, and Robins' 2004 publication establishes a cohesive underlying structural definition of selection bias.[11] Using Directed Acyclic Graphs (DAGs), the authors explore the various biases categorized under the term selection bias and differentiate these biases from bias due to confounding.[11] While various forms of selection bias have been previously identified, Hernán et al. argued that selection bias can be thought of the bias resulting from conditioning on the common effects of two variables, specifically the exposure and outcome or variables that cause the exposure and outcome.[11] The bias resulting from conditioning on common effects, selection bias, thus can be differentiated from the bias resulting from a common cause of the exposure and the outcome, confounding.[11]

The causal model approach to selection bias relies on the phenomenon known as collider-stratification bias, which was elucidated by Greenland in 2003.[12] When variables caused by the exposure and outcome are stratified on or conditioned on, a "back-door pathway" is created that can induce a bias in the causal effect.[12] If this collider is a variable associated with selection or participation in the study, then the bias created is selection bias.

Hernán et al. review the various forms of selection bias and propose appropriate adjustment methods based on the causal structure of each subtype.[11] The paper addresses selection bias among case-control studies due to poor selection of controls, Berkson bias among case-control studies, selection bias from differential Lost to Follow Up (LTFU), non-response/missing data bias,

volunteer/self-selection bias, and healthy worker bias.[11] The authors also draw parallels between the causal structure in these examples and the special case of "adjustment for variables affected by prior exposure".[11]

Though not addressed by Hernán[11], some of these biases can also be categorized under what we would call immigrative selection bias and emigrative selection bias. In general immigrative selection bias refers to selection bias resulting from selection into the study and emigrative selection bias refers to the bias resulting from selection out of the study. Differential LTFU, a notable form of emigrative selection bias, is a common problem in longitudinal studies, including randomized trials[11] Differential LTFU can occur when the treatment or exposure of interest cause side effects resulting in study drop out that is conditional on the exposure level; the outcome for this exposure level cannot be ascertained when individuals are LTFU, resulting in selection bias.[11] Hernán et al. suggest the use of either stratification or inverse probability weighting to correct for differential LTFU.[11] Commonly referred to as inverse probability-of-censoring weights, this procedure has been shown to have some limitations in estimating the causal effect when the sample size is small and/or the magnitude of selection bias is large.[13]

Hernán et al. address several other forms of selection bias that fall under the umbrella term of immigrative selection bias.[11] One such example is Berkson bias, first identified by Berkson in 1946, as the selection bias that can result from using hospital-based controls in case-control studies.[14] Berkson bias can occur when both cases and controls have distinct diseases that increase the likelihood

of hospitalization.[14] A non-causal relationship can be induced between an exposure that causes the disease in controls but not the disease in the cases because selection is conditioned upon being hospitalized.[14] This causal structure fits the structure of selection bias, conditioning on a common effect, proposed by Hernán et al.[11] Volunteer bias and healthy worker bias are similar examples reviewed in the paper.[11] Improved sampling methods and appropriate design through the use of causal structures to identify selection bias mechanisms can help avoid selection bias in these cases.[11] Hernán et al. suggest the use of inverse probability weighting (IPW) for selection bias in general.[11] Weighting based on inverse probability of censoring can address emigrative selection bias, while weighting based on the inverse probability of selection can address immigrative selection bias. A caveat to the latter approach is that it requires information on those not selected into the study in order to determine the conditional probability of selection. This information is often not available unless a particular study is nested within an existing cohort or longitudinal study. This is a major limitation of inverse probability of selection weighting for immigrative selection bias.

A final example addressed by Hernán is the biased from adjustment for variables affected by a previous exposure.[11] When stratification is used to address confounding in this circumstance, selection bias is induced via collider-stratification bias.[11] The proposed alternatives to stratification in this circumstance are the use of “inverse-probability-of-treatment weighting” (IPTW)

or g-estimation if the assumptions of IPTW are violated, e.g. when a conditional probability of treatment is zero.[11]

1.2 Summary and Objectives

As previously discussed, the prognostic score has been found to be an acceptable alternative to the use of the propensity score as a covariate balancing score and can appropriately remove bias due to confounding. There are several methods that can be used to generate the prognostic score. Seemingly the most accepted approach is to estimate the prognostic score among the unexposed group rather than the full study population, given that none of the covariates are effect modifiers. However a recent simulation study revealed comparable results between the two methods, if the assumptions for the full study population prognostic score are met.[3] The prognostic score has also been shown to be a promising supplementary approach to confounding in combination with the propensity score.[5,9]

There are potential applications of the prognostic score to confounding that could prove to be equivalent to the propensity score. Hansen previously highlighted the similarities between the prognostic score and the propensity score, referring to the two balancing scores as analogues.[2] However in the special case of estimating an odds ratio for the relationship between exposure with a binary outcome, either score could theoretically be used and yield the same results because of the properties of the odds ratio. Because the odds ratio of the exposure is equivalent to the odds ratio of the outcome in this case,[15] a balancing score based on the probability of either the outcome or the exposure

as a function of the covariates could be applied to the model and theoretically yield the same odds ratio. That is to say, the exposure or the outcome could serve as the explanatory variable in the model. If the outcome were treated as the explanatory variable, then the propensity score would be the probability of the outcome as a function of the covariates. This “propensity score” is equivalent to the prognostic score that is derived from the full population but one in which the exposure is not set to the unexposed level. The same principle holds if the exposure is treated as the response variable and the prognostic score is calculated using the full population. The result would be the probability of the exposure as a function of the covariates. Note that for it to be equivalent, to estimate the prognostic score would require a slight modification in which the probability of outcome from the full cohort does not include the exposure in the model (Table 1: modified prognostic score).

The equivalency of the prognostic and propensity scores in this case allows for the potential to obtain a doubly robust model using a combination of the two scores as a means to address confounding. One objective of this thesis is to assess the equivalency and performance of the full population prognostic score and the propensity score in a simulation of a confounded logistic model with binary exposure. The simulation will also assess the performance of a dual prognostic score and propensity score approach using combination weights.

The previous section reviews major examples of selection bias and methodological approaches to prevent and account for these biases. While inverse probability-of-censoring weights are an accepted approach for emigrative

selection bias, specifically differential LTFU, methods are more limited for immigrative selection bias. If the study population is nested within a cohort or data is otherwise available on those who opt not to enter the study, then inverse probability-of-selection weights can be used. In the absence of this data, preventative measures can be employed to avoid immigrative selection bias. A potential approach to immigrative selection bias is the use of the prognostic score. Hernán et al. described selection bias as resulting from conditioning on a common effect of the exposure and the outcome or variables that cause the exposure and the outcome.[11] A prognostic score models the probability of the outcome as a function of the covariates. Use of inverse probability weights has been intuitively thought of as removing the relationship between variables included in estimating the weights and the exposure in confounding or for selection into (or out of) the study for selection bias. Using such heuristic logic inverse probability weighting using the prognostic score would correct for selection bias by removing the association between the outcome and the variables inducing the selection bias. A second objective of this thesis to assess the performance of the prognostic score in correcting for immigrative selection bias using simulations that follow the underlying structure of selection bias proposed by Hernán et al.[11] These simulations will emphasize the role of causal models in selection bias and compare the various methodological approaches to selection bias.

Chapter 2: Confounding

2.1 Introduction:

The previous chapter reviewed the development of the prognostic score and its previous application to confounding. Various publications have examined the prognostic score and its derivatives and found that the prognostic score has similar performance to the propensity score.[2-5] This chapter examines the use of several prognostic score approaches including the modified prognostic score in a logistic model in comparison to other approaches including inverse probability of exposure weighting (IPEW). The modified prognostic score models the outcome as a function of the covariates without regard to the exposure status. The modified prognostic score can be thought of as nearly equivalent to the propensity score in a logistic model exploring the relationship between an exposure and a binary outcome. Because the odds of outcome given exposed over the odds of the outcome given unexposed is equal to the odds of exposure given having the outcome over the odds of exposure given not having the outcome,[15] weighting using the modified prognostic score or the propensity score to a logistic model should give equivalent results. Using a DAG with several confounders, this chapter will explore weighting using the modified prognostic score and weighting using the propensity score to address the assumption of their equivalency in a logistic model.

2.2 Methods:

The Monte Carlo simulation in this chapter was based on the DAG presented in Figure 1, which will be termed the Confounding DAG. Figure 1

depicts a binary exposure variable (E), a binary outcome variable (Y), and three binary confounding variables (C1, C2, C3). This simulation compares the effect estimates from various models when the exposure and the outcome have a null association (conditional OR=1) and when the exposure and the outcome have a moderate, positive association (conditional OR=3). Table 2 provides the equations used to generate the variables for the simulation (termed Confounding Simulation).

Each iteration of the simulation generated 1000 observations based on the equations from Table 2. Table 3 describes all of the models used in the Confounding Simulation. Models for the prognostic score included the prognostic score estimated in the unexposed group, the prognostic score estimated in the full sample with an exposure indicator in the model and the exposure status set to zero for the prediction, and a model for the modified prognostic score, where the probability of the outcome (Y) is estimated as a function of the covariates (C1, C2, and C3) without regard for the exposure (E). All of the scores were used to generate stabilized weights to be used in the final regression model. The three prognostic score weights were stabilized by the probability of an individual's observed outcome and compared to inverse probability of exposure stabilized weights, direct adjustment on C1, C2 and C3, and the crude model. The performance of combination propensity and prognostic score methods were also assessed. These combination methods included combining both stabilized IPEW and modified prognostic weights by simply multiplying the weights together, subclassification on propensity score quintiles with weighting using the modified

prognostic score, and subclassification on modified prognostic score quintiles with weighting using the stabilized IPEW. All analyses were performed using R[16] with subclassification methods performed in the MatchIt package.[17]

The marginal odds ratio was determined for each iteration using the method described by Austin in his publication on the performance of the propensity score with respect to estimating the marginal odds ratio.[18] This method requires the calculation of the mean probability of the outcome if all the subjects were exposed to treatment (p_1) and the mean probability of the outcome if all the subjects were not exposed to treatment (p_0).[18] The marginal odds ratio is then estimated by $p_1/(1-p_1)/p_0/(1-p_0)$.[18] The robust variance for each model was calculated in each iteration using the sandwich package in R.[19,20] The percent relative bias and mean squared error (MSE) were calculated for each iteration by comparing the model estimate to both the conditional and marginal odds ratios when OR=3. The robust variance and MSE figures are presented in the results section on the log scale for a better depiction of the distribution of smaller values. Because the marginal and conditional odds ratios are equal when OR=1, these estimates were only compared once for the null model. The Monte Carlo variance for each estimate was determined using the bootstrap estimator. Tables and boxplots of the effect estimates, the robust variance, the percent relative bias, and the MSE were created. To emphasize the utility of the prognostic score approaches and to highlight the equivalency of the modified prognostic score to the propensity score in a logistic model, the simulation models detailed in Table 3 were performed using two approaches. The first approach uses Y as the

response variable and E as the explanatory variable and the second approach uses E as the response variable and Y as the explanatory variable. The purpose of comparing these two approaches is to highlight that the estimation of the odds ratio of exposure comparing those with the outcome to those without the outcome is equivalent to the estimation of the odds ratio of the outcome comparing those with the exposure to those without the exposure. The performance metrics for the crude model and Models 1-6 were determined using both approaches. These models will be referred to as Model 1 (Y) if Y is response variable, Model 1 (E) if E is the response variable and so on.

Figure 1: Confounding DAG

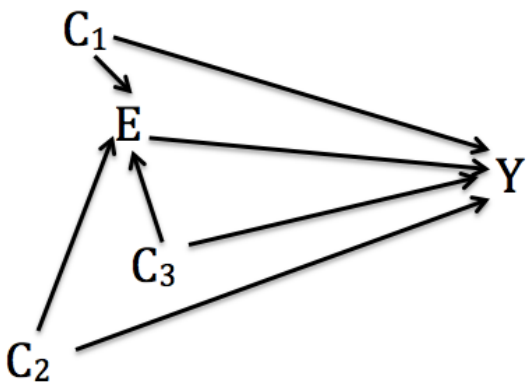


Table 2: Confounding Simulation Equations

Variable	Equations (OR=3)	Equations (OR=1)
Exposure (E)	E~Bin(1000, P(E)) $P(E) = \frac{e^{\left[\log\left(\frac{3}{7}\right)+\log\left(\frac{1}{3}\right)(C1)+\log(3)(C2)+\log\text{^}3(C3)\right]}}{1+e^{\left[\log\left(\frac{3}{7}\right)+\log\left(\frac{1}{3}\right)(C1)+\log(3)(C2)+\log(3)(C3)\right]}}$	E~Bin(1000, P(E)) $P(E) = \frac{e^{\left[\log\left(\frac{3}{7}\right)+\log\left(\frac{1}{3}\right)(C1)+\log(3)(C2)+\log(3)(C3)\right]}}{1+e^{\left[\log\left(\frac{3}{7}\right)+\log\left(\frac{1}{3}\right)(C1)+\log(3)(C2)+\log(3)(C3)\right]}}$
Confounder 1 (C1)	C1~Bin(1000, 0.5)	C1~Bin(1000, 0.5)
Confounder 2 (C2)	C2~Bin(1000, 0.6)	C2~Bin(1000, 0.6)
Confounder 3 (C3)	C3~Bin(1000, 0.5)	C3~Bin(1000, 0.5)
Outcome (Y)	Y~Bin(1000, P(Y)) $P(Y) = \frac{e^{\left[\log\left(\frac{3}{7}\right)+\log(3)(C1)+\log(3)(C2)+\log(3)(C3)+\log(3)(E)\right]}}{1+e^{\left[\log\left(\frac{3}{7}\right)+\log(3)(C1)+\log(3)(C2)+\log(3)(C3)+\log(3)(E)\right]}}$	Y~Bin(1000, P(Y)) $P(Y) = \frac{e^{\left[\log\left(\frac{3}{7}\right)+\log(3)(C1)+\log(3)(C2)+\log(3)(C3)+\log(1)(E)\right]}}{1+e^{\left[\log\left(\frac{3}{7}\right)+\log(3)(C1)+\log(3)(C2)+\log(3)(C3)+\log(1)(E)\right]}}$

Table 3: Confounding Simulation Models

Description	Outcome Regression	Weight Equation	Subclassification
Crude: Crude Model	$\text{Logit}(Y=1)=\beta_0+\beta_1(E)$; $\text{Logit}(E=1)=\beta_0+\beta_1(Y)$	-	-
Model 1: Unexposed Prognostic Weights (Stabilized)	$\text{Logit}(Y=1)=\beta_0+\beta_1(E)$; $\text{Logit}(E=1)=\beta_0+\beta_1(Y)$	$\frac{E[Y=y]}{E[Y=y\|C1=c1,C2=c2,C3=c3,E=0]}$	-
Model 2: Full Sample Prognostic Weights (Stabilized)	$\text{Logit}(Y=1)=\beta_0+\beta_1(E)$; $\text{Logit}(E=1)=\beta_0+\beta_1(Y)$	$\frac{E[Y=y]}{E^{**}[Y=y\|C1=c1,C2=c2,C3=c3,E=e]}$	-
Model 3: Modified Prognostic Weights (Stabilized)	$\text{Logit}(Y=1)=\beta_0+\beta_1(E)$; $\text{Logit}(E=1)=\beta_0+\beta_1(Y)$	$\frac{E[Y=y]}{E[Y=y\|C1=c1,C2=c2,C3=c3]}$	-
Model 4: Stabilized IPEW	$\text{Logit}(Y=1)=\beta_0+\beta_1(E)$; $\text{Logit}(E=1)=\beta_0+\beta_1(Y)$	$\frac{E[E=e]}{E[E=e\|C1=c1,C2=c2,C3=c3]}$	-
Model 5: Direct Adjustment for Confounders	$\text{Logit}(Y=1)=\beta_0+\beta_1(E)+B_2(C1)+\beta_3(C2)+\beta_4(C3)$; $\text{Logit}(E=1)=\beta_0+\beta_1(Y)+B_2(C1)+\beta_3(C2)+\beta_4(C3)$	-	-
Model 6: Combination Weights (Stabilized): IPEW and Modified Prognostic Scores	$\text{Logit}(Y=1)=\beta_0+\beta_1(E)$; $\text{Logit}(E=1)=\beta_0+\beta_1(Y)$	$IPEW=\frac{E[E=e]}{E[E=e\|C1=c1,C2=c2,C3=c3]}$ $ProgW=\frac{E[Y=y]}{E[Y=y\|C1=c1,C2=c2,C3=c3]}$	-
Model 7: Modified Prognostic Weights (Stabilized) and Subclassification on Propensity Score Quintiles	$\text{Logit}(Y=1)=\beta_0+\beta_1(E)$; $\text{Logit}(E=1)=\beta_0+\beta_1(Y)$	$\frac{E[Y=y]}{E[Y=y\|C1=c1,C2=c2,C3=c3]}$	Propensity Score Quintiles $Prop\ Score=\frac{E[E=1]}{E[E=1\|C1=c1,C2=c2,C3=c3]}$
Model 8: Stabilized IPEW and Subclassification on Prognostic Score Quintiles	$\text{Logit}(Y=1)=\beta_0+\beta_1(E)$; $\text{Logit}(E=1)=\beta_0+\beta_1(Y)$	$\frac{E[E=e]}{E[E=e\|C1=c1,C2=c2,C3=c3]}$	Modified Prognostic Score Quintiles $Prog\ Score=\frac{E[Y=1]}{E[Y=1\|C1=c1,C2=c2,C3=c3]}$

Let E**[•]= Pred{E[Y=y|•, E=0]}. (Exposure variable is set to unexposed for prediction of weights).

2.3 Results:

The distribution of the effect estimates for each model of the simulation where the association is moderate (conditional OR=3) are displayed in Figure 2 with reference lines at the conditional and marginal odds ratios. The top left panel in Figure 2 shows results from the Y on E outcome models, while the top right panel shows results from the E on Y outcome models. The lower panels in Figure 2 are the boxplots of the log robust variance estimates for each model with Y as the response variable for the lower left and E as the response variable for the lower right panel. Figure 3 depicts the percent relative bias compared to both the conditional and marginal odds ratios. The top panels display the percent relative bias when Y is the response variable, while the bottom panels display the results when E is the response variable. The left panels in Figure 3 compare the effect estimate to the conditional odds ratio while the right panels compare the estimate to the marginal odds ratio. Figure 4 shows the log MSE based on both the conditional and marginal ORs using Y then E as the response variable. Table 4 provides a summary of the results stratified by the response variable for the Confounding Simulation where the conditional odds ratio is 3.

Based on Figures 2-4 it appears that for all models in the simulation, the effect estimates, robust variance estimates, percent relative bias, and MSE appear to be nearly equivalent regardless of whether the model uses Y or E as the response variable. This is as expected given previous knowledge about the logistic model and the odds ratio. Because the performance metrics for all of the models are equivalent regardless of the response variable except for Model 5,

which directly adjusts for the confounders, Table 4 lists the results for the two response variables in one row for all other models. The results for the direct adjustment model, Model 5, differ slightly based on which response variable was used; however, the effect estimates and 95% bootstrap confidence intervals are quite similar. The estimated logOR for Model 5 is 1.1058 with a 95% CI of [0.72, 1.33] when Y is the response variable and 1.0997 with a 95% CI of [0.75, 1.47] when E is the response variable. The mean marginal odds ratio across the 1,000 iterations was found to be 2.597 (logOR=0.954).

Models 3 (modified prognostic score) and 4 (IPEW) appear to best estimate the marginal odds ratio, while the other prognostic scores, direct adjustment, and the combined prognostic and propensity score approaches (Models 1, 2, and 5-8) appear to approximate the conditional odds ratio. The median relative percent bias comparing the modified prognostic score weights, Model 3, and stabilized IPEW, Model 4, to the marginal odds ratio is -0.4693% and 0.7258%, respectively. Conversely, Models 3 and 4 have a median relative bias of approximately -13% when compared to the conditional OR. Models 1 (unexposed prognostic score) and 2 (full sample prognostic score) have a 0.2478% and 0.9973% median percent relative bias when compared to the conditional odds ratio, while the median percent relative bias is 15.4% and 16.4%, respectively, when compared to the marginal odds ratio. The direct adjustment approach (Model 5) has a 0.66% median relative bias compared to the conditional OR when Y is the response versus 0.1% when E is the response variable, while the bias compared to the marginal OR is 16% for Y as the

response variable and 15% for E as the response variable.

Interestingly, the various combined modified prognostic and propensity score approaches yield an estimate closer to the conditional than the marginal OR even though the individual modified and IPEW approaches estimated the marginal OR. When compared to the conditional OR for both response variables, Model 6 has 0.67% median relative bias, Model 7 has -1.19% median relative bias, and Model 8 has -1.32% median relative bias.

The Monte Carlo variance estimates and median robust variance estimates are similar for all of the models on with values ranging from 0.027 to 0.048. The estimates for MSE are also similar between the models with values ranging from 0.04 to 0.086. Apart from the crude model, IPEW and direct adjustment, Model 4 and Model 5, have the lowest bootstrap and robust variances. Among the prognostic score approaches (Models 1-3), the modified prognostic score has the lowest Monte Carlo variance while the full sample prognostic score has the lowest robust variance estimate. Compared to the conditional OR, the crude model has the lowest MSE while direct adjustment has the second lowest MSE. Compared to the marginal OR, the IPEW model has the lowest MSE, while the modified prognostic score model has the second lowest MSE.

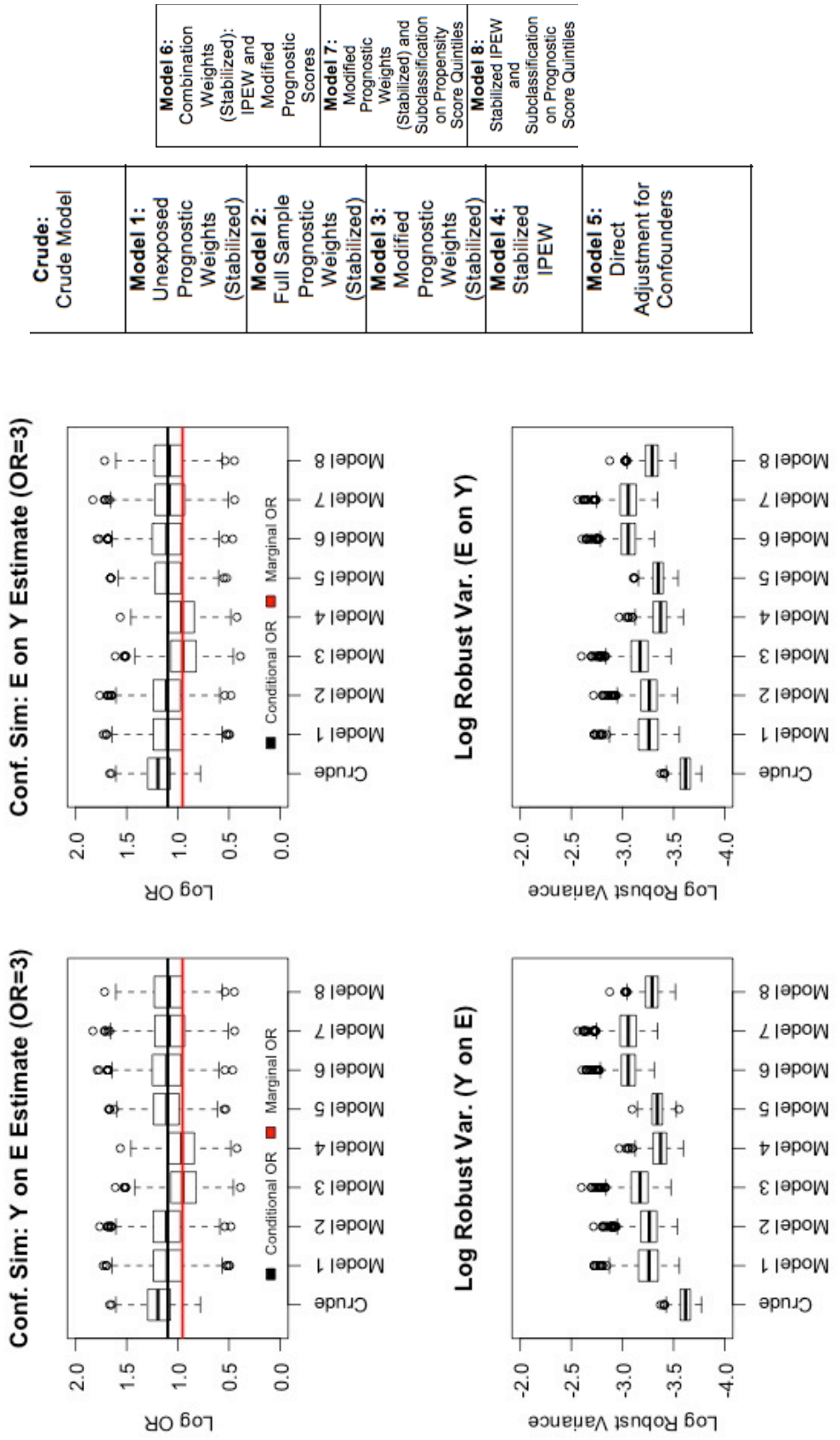


Figure 2: Confounding Simulation Effect Estimates and Log Robust Variance (Conditional OR=3)

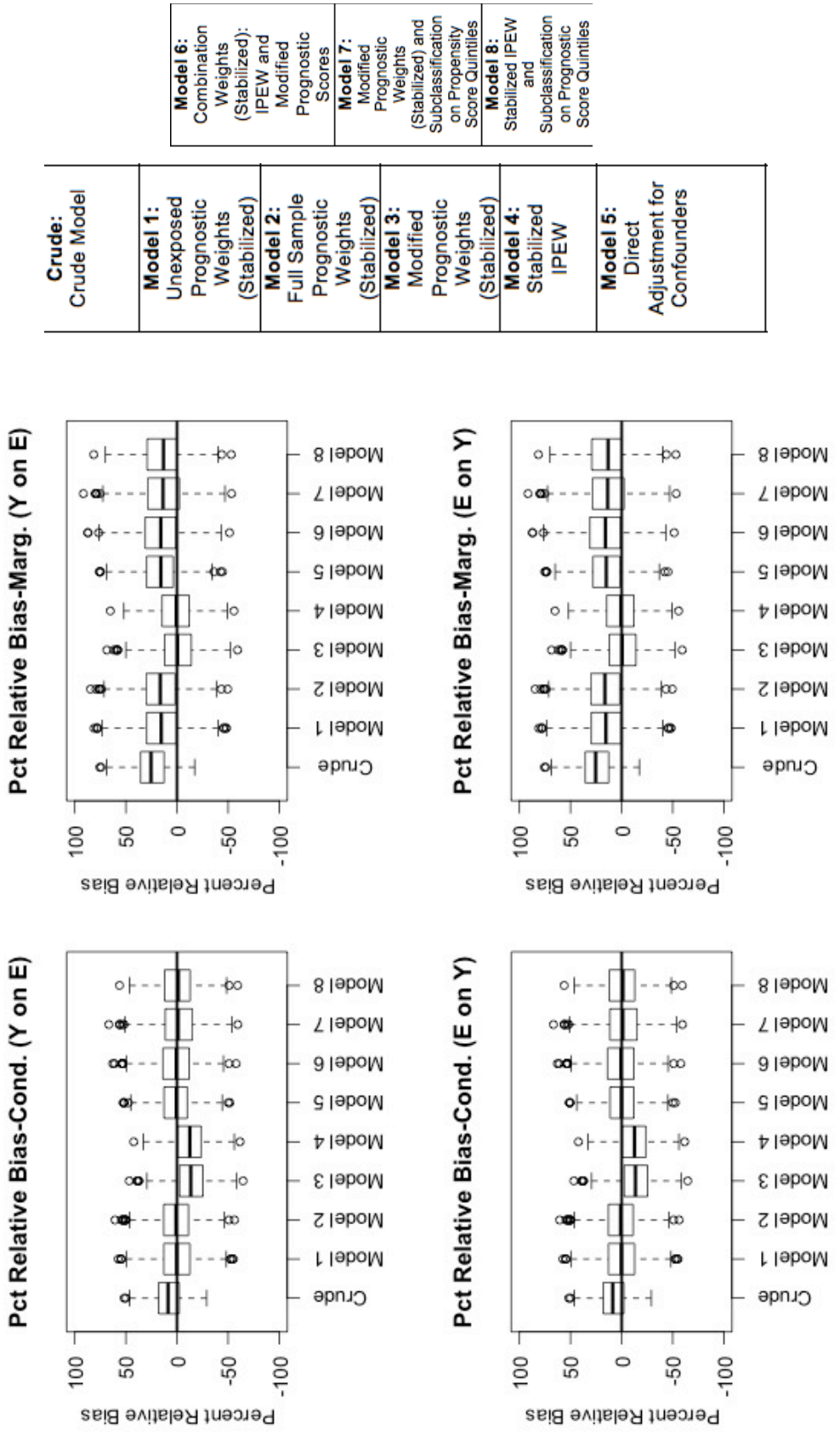


Figure 3: Confounding Simulation (Conditional OR=3) Percent Relative Bias

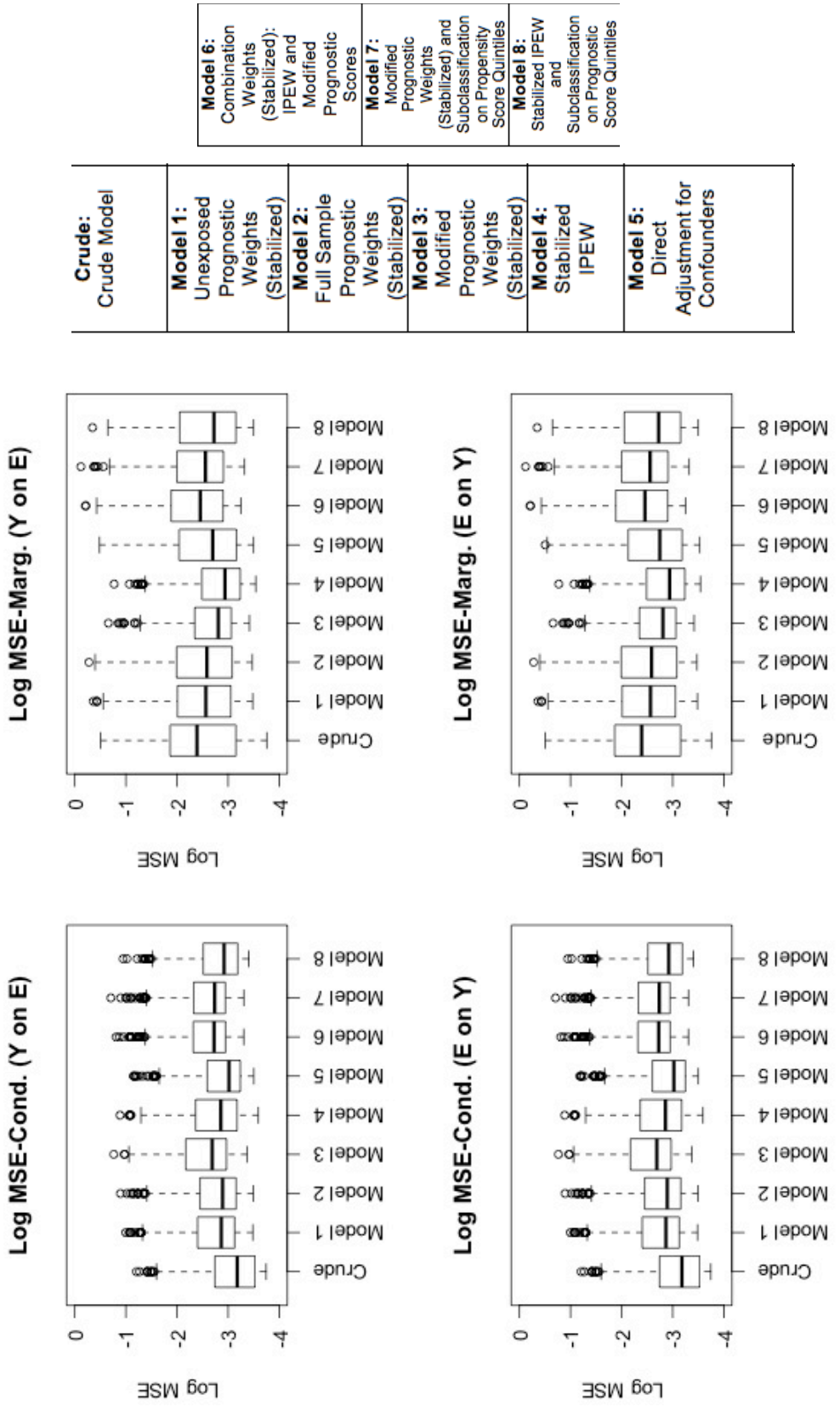


Figure 4: Confounding Simulation (Conditional OR=3) Log Mean Squared Error

Table 4: Results for Confounding Simulation, OR=3

	Outcome Variable	Median logOR [95% CI Bootstrap]	Median Robust Variance	Median % Relative Bias (Conditional)	Median % Relative Bias (Marginal)	Median MSE (Conditional)	Median MSE (Marginal)	Monte Carlo Variance (bootstrap)
Crude Model	Y or E	1.1946 [0.89, 1.52]	0.02686	8.736	25.43	0.04160	0.09149	0.02652
Model 1: Unexposed Prognostic Weights (Stabilized)	Y or E	1.1013 [0.70, 1.52]	0.03843	0.2478	15.401	0.05687	0.07691	0.04465
Model 2: Full Sample Prognostic Weights (Stabilized)	Y or E	1.1096 [0.71, 1.54]	0.03828	0.9973	16.366	0.05539	0.07534	0.04269
Model 3: Modified Prognostic Weights (Stabilized)	Y or E	0.9503 [0.59, 1.36]	0.04191	-13.498	-0.4693	0.06797	0.06028	0.03689
Model 4: IPEW	Y or E	0.9605 [0.72, 1.33]	0.03429	-12.572	0.7258	0.05736	0.05289	0.03238
Model 5: Direct Adjust	Y	1.1058 [0.63, 1.31]	0.03540	0.6565	15.829	0.04890	0.06727	0.03536
Model 5: Direct Adjust	E	1.0997 [0.75, 1.47]	0.03509	0.1009	15.058	0.04870	0.06408	0.03532
Model 6: Combination Weights: Modified Prognostic Score and IPEW	Y or E	1.1059 [0.76, 1.45]	0.04705	0.6674	15.896	0.06553	0.08577	0.04786
Model 7: Mod. Prog Weights, Subclassification by Propensity Score	Y or E	1.0855 [0.72, 1.58]	0.04705	-1.192	13.663	0.06472	0.07744	0.04652
Model 8: IPEW, Subclassification by Mod. Prog Score	Y or E	1.0841 [0.68, 1.55]	0.03725	-1.320	13.249	0.05380	0.06564	0.03777

The distribution of the effect estimate and log robust variance for each model of the simulation where there is a null association (conditional OR=1) is displayed in Figure 5. The top panels of Figure 5 were generated using Y (left) and E (right) as the response variable with a reference line added to represent the conditional/marginal odds ratio, which are equivalent in the null model. The bottom panels of Figure 5 are the robust variance estimates for each model with Y (left) as the response variable and E (right) as the response variable.

Figure 6 is the log MSE when the conditional OR is equal to 1 when the response variable is set to Y (top panel) and when the response variable is set to E (bottom panel). Table 5 provides a summary of the results for the Confounding Simulation with a null effect estimate. As indicated in the figures, the distribution of the effect estimate, the log robust variance, and the log MSE are essentially equivalent regardless of which response variable is used. The direct adjustment model (Model 5) has slightly different results, which are presented in two rows of Table 5. Otherwise, Table 5 lists the results for the two response variables in one row for the other models. The estimated logOR for Model 5 is 0.0004 with a 95% CI of [-0.30, 0.32] when Y is the response variable and -0.0037 with a 95% CI of [-0.30, 0.32] when E is the response variable. The mean marginal odds ratio across the 1,000 iterations was found to be 1.000 (logOR=0). Given that the marginal and conditional odds ratios are equivalent with the null effect estimate, the observed differences in marginal versus conditional ORs among the models when the odds ratio was moderate (OR=3) are not present. Models 1 through 6 appropriately account for confounding and yield a median effect estimate similar

to that of the conditional/marginal estimate. The median log OR for the crude model is 0.2126 while the median log OR for Models 1-8 range between -0.0092 and 0.014.

The Monte Carlo variance estimates and median robust variance estimates are similar for all of the models with most values in the 0.02 to 0.03 range. The estimates for MSE are also similar between the models with values ranging from 0.034 to 0.064. Apart from the crude model, IPEW, Model 4, has the lowest bootstrap and robust variances. Among the prognostic score approaches (Models 1-3), the modified prognostic score has the lowest Monte Carlo variance and robust variance, though the full sample prognostic score has a similar robust variance. Model 4, IPEW, has the lowest MSE.

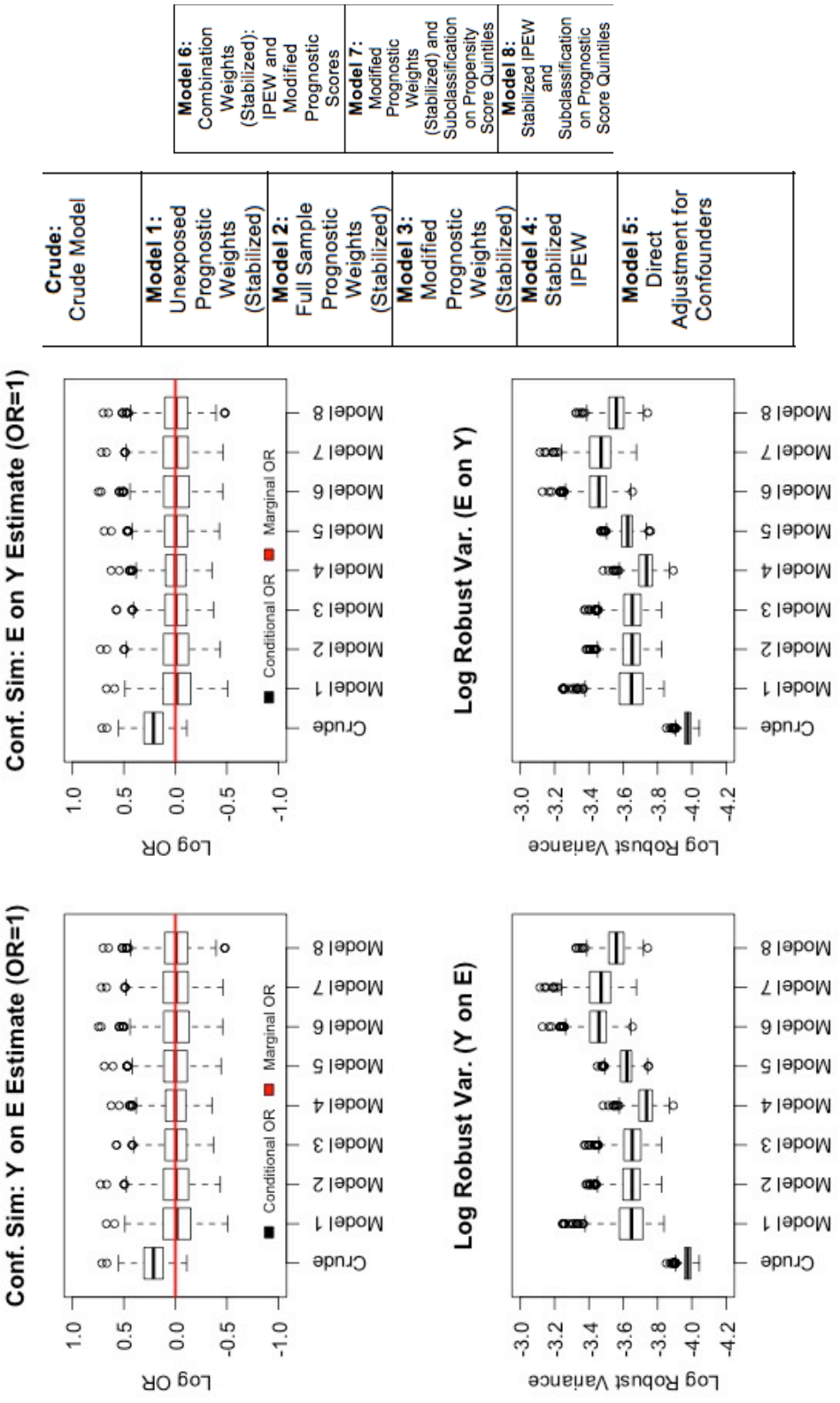
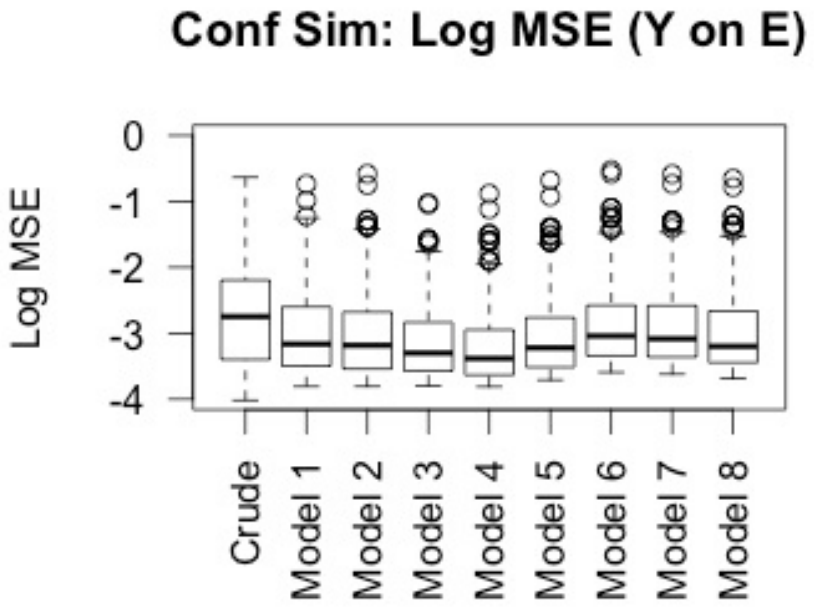


Figure 5: Confounding Simulation Effect Estimates and Log Robust Variance (OR=1)

Figure 6: Confounding Simulation (Conditional OR=1)
Log Mean Squared Error



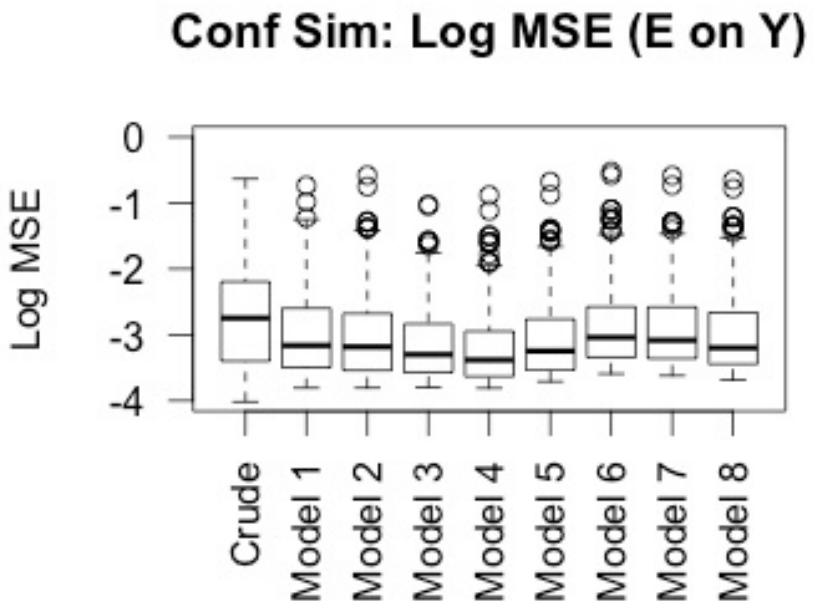


Table 5: Results for Confounding Simulation, OR=1

	Response Variable	Median logOR [95% CI Bootstrap]	Median Robust Variance	Median MSE (Conditional)	Monte Carlo Variance (bootstrap)
Crude Model	Y or E	0.2126 [-0.035, 0.49]	0.01877	0.06413	0.01784
Model 1: Unexposed Prognostic Weights (Stabilized)	Y or E	-0.014 [-0.37, 0.36]	0.02605	0.04228	0.03518
Model 2: Full Sample Prognostic Weights (Stabilized)	Y or E	-0.0080 [-0.33, 0.38]	0.02594	0.04140	0.03135
Model 3: Modified Prognostic Weights (Stabilized)	Y or E	-0.0066 [-0.29, 0.34]	0.02593	0.03684	0.02319
Model 4: IPEW	Y or E	-0.0005 [-0.27, 0.28]	0.02385	0.03396	0.02098
Model 5: Direct Adjust	Y	0.0004 [-0.30, 0.32]	0.02675	0.03987	0.02716
Model 5: Direct Adjust	E	-0.0037 [-0.30, 0.32]	0.02662	0.03892	0.02706
Model 6: Combination Weights: Modified Prognostic Score and IPEW	Y or E	-0.0036 [-0.31, 0.41]	0.03149	0.04778	0.0334
Model 7: Mod. Prog Weights, Subclassification by Propensity Score	Y or E	-0.0092 [-0.31, 0.37]	0.03109	0.04578	0.03229
Model 8: IPEW, Subclassification by Mod. Prog Score	Y or E	-0.0077 [-0.32, 0.32]	0.02844	0.04084	0.02869

2.4 Discussion:

This chapter examined the use of the prognostic score in a logistic model and a special case of the prognostic score termed the modified prognostic score, where the outcome was modeled as a function of the covariates without regard to exposure status. It was hypothesized that the modified prognostic score would function in an equivalent manner to the propensity score in a causal model that is subject to confounding. The results of the simulation indicated that weighting using any version of the prognostic score or weighting using the propensity score removed bias due to confounding. It was found that the modified prognostic score and IPEW yielded the marginal odds ratio when the effect estimate was moderate (conditional OR=3), while other variants of the prognostic score and direct adjustment on confounders resulted in the conditional odds ratio. It was presumed that the logistic model used in Chapter 2 was a special case where the modified prognostic and propensity score would be equivalent and the response variable could be either the outcome or the exposure. As observed in the results, the effect estimate was the same regardless of the response variable, and the prognostic and propensity scores yield similar effect estimates.

Another objective of this chapter was to assess the performance of different methods combining the modified prognostic score and the propensity score. Combination weights using the two scores, weighting using the modified prognostic score with subclassification on the propensity score quintiles, and weighting using the propensity score with subclassification on the modified prognostic score quintiles were compared. While weighting on the modified

prognostic or propensity score yielded the marginal odds ratio, the combination methods yielded estimates closer to the conditional odds ratio. This was a surprising result, because it was assumed that the combination methods would result in a marginal estimate similar to the results for the individual components of the combination methods.

The modified prognostic score will be further examined in Chapter 3 in an application to selection bias; however, more information regarding the performance of the modified prognostic score is needed. Additionally, information regarding the use of weighting in combined prognostic and propensity score methods is needed.

Chapter 3: Selection Bias

3.1 Introduction

Chapter 1 introduces the background of the prognostic score and its potential application to selection bias. The prognostic score has been applied to confounding though not to the extent of its balancing score counterpart, the propensity score.[6] This chapter extends the use of the prognostic score to selection bias. Intuitively, if the conceptual frame of inverse probability weights is to remove the arrow in a directed acyclic graph between a confounder and exposure for confounding or similarly for selection bias between exposure and other variables that cause selection, then utilizing prognostic scores as a weight may also remove selection bias. If this is so then the prognostic score represents a potential methodological approach to immigrative selection bias, where current methods require information about those not selected into the study in order to estimate the conditional probability of selection.

The prognostic score can be derived using several methods, which will be termed the full population prognostic score, the unexposed prognostic score, and the modified prognostic score in this chapter. The full population prognostic score estimates the probability of the outcome as a function of the covariates including the exposure among the entire population and this model is used to generate a predicted score in which the exposure variable set to zero.[1] The score generated by this procedure is often referred to as the full cohort disease risk score and is discussed further in Chapter 1.[3] The unexposed prognostic score, also termed the unexposed-only disease risk score, estimates the probability of the outcome

as a function of the covariates among those in the study population who are unexposed. Then a score is predicted for the entire population (both exposed and unexposed) based on this model.[2] Finally, the modified prognostic score estimates the probability of the outcome as a function of the covariates without regard to exposure status.

Stuart et al. delineated that the term prognostic score refers to the extension of the disease risk score beyond binary outcomes to continuous, categorical, and ordinal outcomes.[5] While Stuart et al. and Hansen argue for the generation of the prognostic score among the unexposed group, Arbogast et al. has shown that the full population score performs well in the event that the additional assumptions are met.[3] Chapter 2 reviewed the use of the modified prognostic score in confounding. Using DAGs, this chapter will explore the performance of the three prognostic score variants compared to existing methods to correct for selection bias and both selection bias and confounding.

3.2 Methods

In this chapter, weighting approaches using the various prognostic scores are compared to inverse probability-of-selection weights (IPSW), to combination weighting approaches using prognostic scores and IPSW, and to direct adjustment for variables that induce selection using Monte Carlo simulations based on several DAGs. Though weighting is an accepted approach with the propensity score,[21,22] it has not been significantly explored using the prognostic score.[6,9]

The Monte Carlo simulations in this chapter were based on three different

DAGs that include selection bias or selection bias and confounding. Selection DAG 1 (Figure 7) depicts a binary exposure variable (E), a binary outcome variable (Y), a variable representing selection into the study (S), and a variable that affects selection into the study (L). The variable representing selection (S) into the study will be referred to as the selection variable and the variable that affects selection into the study (L) will be referred to as a selection covariate for the purposes of this simulation. This DAG follows the causal structure for selection bias proposed by Hernán et al.[11] because the exposure (E) and the selection covariate (L) that causes the outcome (Y) collide at the selection variable (S). Only those who are selected into the study can be included in the analysis, so by default the selection variable (S) is always conditioned on, and in this case it is a collider that meets the structural criteria for selection bias. Simulation 1, based on Selection DAG 1, explores regression results when the exposure and the outcome have a null association (conditional OR=1) and when the exposure and the outcome have a moderate, positive association (conditional OR=3). Table 6 provides the equations used to generate the variables for Simulation 1.

For Simulation 1, each iteration generated a source population of 1000 observations based on the equations from Table 6. Observations where S=1 were included in the study population (i.e. selected into the study). Several models were built based on unexposed prognostic score, the full population prognostic score, the modified prognostic score, IPSW, and direct adjustment. These models were compared with each other, to the crude model, and to the

model results from those selected and unselected into the study. Table 7 describes all of the models used in Simulation 1. The marginal odds ratio for Simulation 1 was determined using Austin's method as in Chapter 2.[18]

A total of 1,000 iterations of Simulation 1 were performed. The robust variance for each model was calculated in each iteration using the sandwich package in R.[19,20] As in Chapter 2, the percent relative bias and mean squared error (MSE) were calculated for each iteration by comparing the model estimate to both the conditional and marginal odds ratios when the conditional OR=3. The effect estimates (log OR) and robust variance was also obtained for each iteration. The Monte Carlo variance for each estimate was determined using the bootstrap estimator after 1,000 iterations. When the conditional OR=1, the effect estimates (log OR), robust variance, MSE, and Monte Carlo variance were calculated. Boxplots of the effect estimates, the log robust variance, the percent relative bias compared to both the Marginal and Conditional log ORs, and the log MSE of the estimate compared to the Marginal and Conditional log ORs were created. All analyses were performed in R.[16]

Figure 7: Selection DAG 1

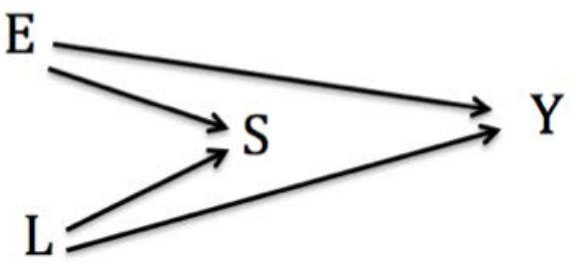


Table 6: Selection Simulation 1 Equations

Variable	Equation (OR=3)	Equation (OR=1)
Exposure (E)	E~ Bin(1000, 0.3)	E~ Bin(1000, 0.3)
Selection Covariate (L)	L~Bin(1000, 0.6)	L~Bin(1000, 0.6)
Selection Variable (S)	S~Bin(1000, P(S)) $P(S) = \frac{e^{\left[\log\left(\frac{.6}{.4}\right)+\log(6)(E)+\log(5)(L)\right]}}{1+e^{\left[\log\left(\frac{.6}{.4}\right)+\log(6)(E)+\log(5)(L)\right]}}$	S~Bin(1000, P(S)) $P(S) = \frac{e^{\left[\log\left(\frac{.6}{.4}\right)+\log(6)(E)+\log(5)(L)\right]}}{1+e^{\left[\log\left(\frac{.6}{.4}\right)+\log(6)(E)+\log(5)(L)\right]}}$
Outcome (Y)	Y~Bin(1000, P(Y)) $P(Y) = \frac{e^{\left[\log\left(\frac{.3}{.7}\right)+\log(3)(E)+\log(5)(L)\right]}}{1+e^{\left[\log\left(\frac{.3}{.7}\right)+\log(3)(E)+\log(5)(L)\right]}}$	Y~Bin(1000, P(Y)) $P(Y) = \frac{e^{\left[\log\left(\frac{.3}{.7}\right)+\log(1)(E)+\log(5)(L)\right]}}{1+e^{\left[\log\left(\frac{.3}{.7}\right)+\log(1)(E)+\log(5)(L)\right]}}$

Table 7: Selection Simulation 1 Models

Description	Formula	Weight Equation	Data Set in Model
Crude: Crude Model in Sample Population	$\text{Logit}(Y=1)=\beta_0+\beta_1(E)$	-	Selected Sample Population (S=1)
Model 1: Unexposed Prognostic Score Weights (Stabilized)	$\text{Logit}(Y=1)=\beta_0+\beta_1(E)$	$\frac{E[Y=y]}{E[Y=y\|L=l,E=0]}$	Selected Sample Population (S=1)
Model 2: Full Sample Prognostic Score Weights (Stabilized)	$\text{Logit}(Y=1)=\beta_0+\beta_1(E)$	$\frac{E[Y=y]}{E^{**}[Y=y\|L=l,E=e]}$	Selected Sample Population (S=1)
Model 3: Modified Prognostic Score Weights (Stabilized)	$\text{Logit}(Y=1)=\beta_0+\beta_1(E)$	$\frac{E[Y=y]}{E[Y=y\|L=l]}$	Selected Sample Population (S=1)
Model 4: Stabilized IPSW (Inverse Probability of Selection Weights)	$\text{Logit}(Y=1)=\beta_0+\beta_1(E)$	$\frac{E[S=s]}{E[S=s\|L=l,E=e]}$	Selected Sample Population (S=1)
Model 5: Direct Adjustment on Selection Covariate (L)	$\text{Logit}(Y=1)=\beta_0+\beta_1(E)+B_2(L)$	-	Selected Sample Population (S=1)
Model 6: Crude Model in Full Population	$\text{Logit}(Y=1)=\beta_0+\beta_1(E)$	-	Full Population (S=1, S=0)

Let E**[•]= Pred{E[Y=y|•, E=0]}. (Exposure variable is set to unexposed for prediction of weights).

Selection DAG 2 (Figure 8) builds upon Selection DAG 1 by introducing a confounder into the selection bias causal model. The purpose of this simulation is to examine the performance of various prognostic score approaches, including combination weights using IPEW, in situations of selection bias and confounding. Similar to Simulation 1, Simulation 2 is based on Selection DAG 2 and explores the results for a null relationship (conditional OR=1) and a moderate, positive association (conditional OR=3). Table 8 provides the equations for Simulation 2.

Each iteration of Simulation 2 generated 1,000 observations. Observations where S=1 were included in the study population (i.e. selected into the study). A total of 1,000 iterations of Simulations 2 were performed. Several models were built using unexposed prognostic score weights based on L and C, the full population prognostic score weights based on L, C, and E (setting exposure to unexposed when estimating the prognostic score weight), the modified prognostic score weights based on L and C, combination weights using IPW based on C and each variant of the prognostic score based on L only. Other models employed direct adjustment for L and C, direct adjustment for L only, and direct adjustment for C. The crude model and the model based on those selected and unselected into the study were also included in the simulation.

Table 9 lists all of the models used in Simulation 2. As seen in Table 9, models 1-7 are the different approaches to control for both selection bias and confounding, while models 8-11 only partially address these biases. The marginal odds ratio for Simulation 2 was determined using the same method as in Simulation 1. As in Simulation 1, the log robust variance and effect estimates

were compared. The MSE were compared between models, and information was presented regarding the percent relative bias compared to the conditional and marginal odds ratios when the conditional OR=3. The Monte Carlo variance for each estimate was determined using the bootstrap estimator after 1,000 iterations. Boxplots of the comparisons were also created.

Figure 8: Selection DAG 2

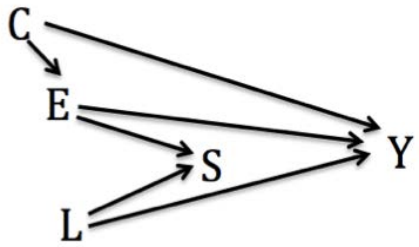


Table 8: Selection Simulation 2 Equations

Variable	Equations (OR=3)	Equations (OR=1)
Exposure (E)	E~ Bin(1000, P(E)) $P(E) = \frac{e^{\left[\log\left(\frac{6}{4}\right)+\log(1/3)(C)\right]}}{1+e^{\left[\log\left(\frac{6}{4}\right)+\log(1/3)(C)\right]}}$	E~ Bin(1000, P(E)) $P(E) = \frac{e^{\left[\log\left(\frac{6}{4}\right)+\log(1/3)(C)\right]}}{1+e^{\left[\log\left(\frac{6}{4}\right)+\log(1/3)(C)\right]}}$
Confounder (C)	C~ Bin(1000, 0.3)	C~ Bin(1000, 0.3)
Selection Covariate (L)	L~Bin(1000, 0.6)	L~Bin(1000, 0.6)
Selection Variable (S)	S~Bin(1000, P(S)) $P(S) = \frac{e^{\left[\log\left(\frac{6}{4}\right)+\log(6)(E)+\log(5)(L)\right]}}{1+e^{\left[\log\left(\frac{6}{4}\right)+\log(6)(E)+\log(5)(L)\right]}}$	S~Bin(1000, P(S)) $P(S) = \frac{e^{\left[\log\left(\frac{6}{4}\right)+\log(6)(E)+\log(5)(L)\right]}}{1+e^{\left[\log\left(\frac{6}{4}\right)+\log(6)(E)+\log(5)(L)\right]}}$
Outcome (Y)	Y~Bin(1000, P(Y)) $P(Y) = \frac{e^{\left[\log\left(\frac{3}{7}\right)+\log(2)(C)+\log(5)(L)+\log(3)(E)\right]}}{1+e^{\left[\log\left(\frac{3}{7}\right)+\log(2)(C)+\log(5)(L)+\log(3)(E)\right]}}$	E~Bin(1000, P(E)) $P(Y) = \frac{e^{\left[\log\left(\frac{3}{7}\right)+\log(2)(C)+\log(5)(L)+\log(1)(E)\right]}}{1+e^{\left[\log\left(\frac{3}{7}\right)+\log(2)(C)+\log(5)(L)+\log(1)(E)\right]}}$

Table 9: Selection Simulation 2 Models

Description	Equation	Weight Equation	Data Set
Crude: Crude Model	$\text{Logit}(Y=1)=\beta_0+\beta_1(E)$	-	Selected Sample Population (S=1)
Model 1: Unexposed Prog. Weights using L and C (Stabilized)	$\text{Logit}(Y=1)=\beta_0+\beta_1(E)$	$\frac{E[Y=y]}{E[Y=y\|L=l,C=c,E=0]}$	Selected Sample Population (S=1)
Model 2: Full Sample Prog. Weights using L and C (Stabilized)	$\text{Logit}(Y=1)=\beta_0+\beta_1(E)$	$\frac{E[Y=y]}{E^{**}[Y=y\|L=l,C=c,E=e]}$	Selected Sample Population (S=1)
Model 3: Modified Prog. Weights using L and C (Stabilized)	$\text{Logit}(Y=1)=\beta_0+\beta_1(E)$	$\frac{E[Y=y]}{E[Y=y\|L=l,C=c]}$	Selected Sample Population (S=1)
Model 4: Combination Weights (Stabilized): IPEW using C; Unexposed Prognostic Weights using L	$\text{Logit}(Y=1)=\beta_0+\beta_1(E)$	$UnexpProgW=\frac{E[Y=y]}{E[Y=y\|L=l,E=0]}$ $IPEW=\frac{E[E=E]}{E[E=e\|C=c]}$	Selected Sample Population (S=1)
Model 5: Combination Weights (Stabilized): IPEW using C; Full Sample Prognostic Weights using L	$\text{Logit}(Y=1)=\beta_0+\beta_1(E)$	$FullProgW=\frac{E[Y=y]}{E[Y=y,E=0\|L=l,E=e]}$ $IPEW=\frac{E[E=E]}{E[E=e\|C=c]}$	Selected Sample Population (S=1)
Model 6: Combination Weights (Stabilized): IPW using C; Modified Prognostic Weights using L	$\text{Logit}(Y=1)=\beta_0+\beta_1(E)$	$ModProgW=\frac{E[Y=y]}{E[Y=y\|L=l]}$ $IPEW=\frac{E[E=E]}{E[E=e\|C=c]}$	Selected Sample Population (S=1)
Model 7: Direct Adjustment for L and C	$\text{Logit}(Y=1)=\beta_0+\beta_1(E)+B_2(L)+B_3(C)$	-	Selected Sample Population (S=1)
Model 8: Direct Adjustment for C among Full Pop. (No Selection)	$\text{Logit}(Y=1)=\beta_0+\beta_1(E)+B_2(C)$	-	Full Population (S=1, S=0)
Model 9: Crude Model among Full Pop. (Confounded)	$\text{Logit}(Y=1)=\beta_0+\beta_1(E)$	-	Full Population (S=1, S=0)
Model 10: Direct Adjustment for C among Sample Pop. (Selection Bias)	$\text{Logit}(Y=1)=\beta_0+\beta_1(E)+B_2(C)$	-	Selected Sample Population (S=1)
Model 11: Direct Adjustment for L among Sample Pop. (Confounded)	$\text{Logit}(Y=1)=\beta_0+\beta_1(E)+B_2(L)$	-	Selected Sample Population (S=1)

Let E**[•]= Pred{E[Y=y|•, E=0]}. (Exposure variable is set to unexposed for prediction of weights).

Selection DAG 3 (Figure 9) includes the exposure variable (E), selection covariate (L), selection variable (S), and outcome variable (Y). However, it introduces an additional unmeasured selection covariate (U) and L is now a collider with S as a descendent node. In this simulation, methods to address selection bias are limited. Because S is by default conditioned on when only selected populations are included in the analysis, L would be partially conditioned on given that it is upstream of S, resulting in selection bias.

Adjusting for L would likely increase the collider-stratification bias in the estimate. Given that U is unmeasured, traditional approaches require information on those not selected into the study or would not account for the effect of selection bias. If both those selected (S=1) and those not selected (S=0) into the sample provided information on L, then IPSW could be used to remove selection bias. However, with immigrative selection bias this usually is not the situation and L is unknown among those not included in the sample. This is a major limitation of the IPSW approach as well as other approaches to address selection bias. Similar to previous simulations, Simulation 3 is based on Selection DAG 3 and explores the results for a null relationship (conditional OR=1) and a moderate, positive association (conditional OR=3). Table 10 provides the equations used to generate Simulation 3.

Simulation 3 generated 1,000 observations and a total of 1,000 iterations were performed. Observations where S=1 were included in the study population (i.e. selected into the study). Table 11 lists all of the models used in Simulation 3. Models 1-5 use the measured data and were built using unexposed prognostic

score weights based on L, the full population prognostic score weights based on L, the modified prognostic score weights based on L, the IPSW based on E and L, and direct adjustment on L in addition to the crude model. Models 6-10 use the various prognostic scores, IPSW, and direct adjustment on L and U, although U would be unavailable in practice. Model 11 based on those selected and unselected into the study was also included in the simulation.

As in Simulation 1 and 2, the marginal odds ratio, robust variance, effect estimates, and MSE were calculated for each iteration. Information was presented regarding the percent relative bias compared to the conditional and marginal odds ratios when the conditional OR was set to 3. The Monte Carlo variance in Simulation 3 for each model was determined using the bootstrap.

Figure 9: Selection DAG 3

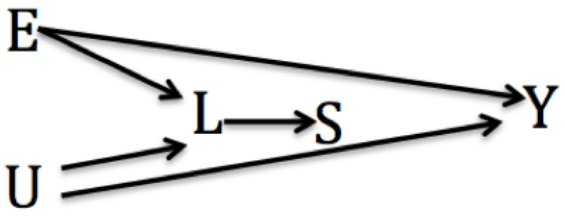


Table 10: Selection Simulation 3 Equations

Variable	Equation	Equation
Exposure (E)	E~ Bin(1000, 0.3)	E~ Bin(1000, 0.3)
Unmeasured Selection Covariate (U)	U~ Bin(1000, 0.3)	U~ Bin(1000, 0.3)
Selection Covariate (L)	L~Bin(1000, P(L)) $P(L) = \frac{e^{\left[\log\left(\frac{.3}{.7}\right)+\log(6)(E)+\log(6)(U)\right]}}{1+e^{\left[\log\left(\frac{.3}{.7}\right)+\log(6)(E)+\log(6)(U)\right]}}$	L~Bin(1000, P(L)) $P(L) = \frac{e^{\left[\log\left(\frac{.3}{.7}\right)+\log(6)(E)+\log(6)(U)\right]}}{1+e^{\left[\log\left(\frac{.3}{.7}\right)+\log(6)(E)+\log(6)(U)\right]}}$
Selection Variable (S)	S~Bin(1000, P(S)) $P(S) = \frac{e^{\left[\log\left(\frac{.3}{.7}\right)+\log(6)(L)\right]}}{1+e^{\left[\log\left(\frac{.3}{.7}\right)+\log(6)(L)\right]}}$	S~Bin(1000, P(S)) $P(S) = \frac{e^{\left[\log\left(\frac{.3}{.7}\right)+\log(6)(L)\right]}}{1+e^{\left[\log\left(\frac{.3}{.7}\right)+\log(6)(L)\right]}}$
Outcome (Y)	Y~Bin(1000, P(Y)) $P(Y) = \frac{e^{\left[\log\left(\frac{.3}{.7}\right)+\log(3)(E)+\log(6)(U)\right]}}{1+e^{\left[\log\left(\frac{.3}{.7}\right)+\log(3)(E)+\log(6)(U)\right]}}$	E~Bin(1000, P(E)) $P(Y) = \frac{e^{\left[\log\left(\frac{.3}{.7}\right)+\log(3)(E)+\log(6)(U)\right]}}{1+e^{\left[\log\left(\frac{.3}{.7}\right)+\log(3)(E)+\log(6)(U)\right]}}$

Table 11: Selection Simulation 3 Models

Description	Equation	Weights	Data Set
Crude: Crude Model	Logit(Y=1)=β_0 + β_1(E)	-	Selected Sample Population (S=1)
Model 1: Unexposed Prognostic Weights using L (Stabilized)	Logit(Y=1)=β_0 + β_1(E)	$\frac{E[Y=y]}{E[Y=y\|L=l,E=0]}$	Selected Sample Population (S=1)
Model 2: Full Sample Prognostic Weights using L (Stabilized)	Logit(Y=1)=β_0 + β_1(E)	$\frac{E[Y=y]}{E^{**}[Y=y\|L=l,E=e]}$	Selected Sample Population (S=1)
Model 3: Modified Prognostic Weights using L (Stabilized)	Logit(Y=1)=β_0 + β_1(E)	$\frac{E[Y=y]}{E[Y=y\|L=l]}$	Selected Sample Population (S=1)
Model 4: Stabilized IPSW using E and L	Logit(Y=1)=β_0 + β_1(E)	$\frac{E[S=s]}{E[S=s\|L=l,E=e]}$	Selected Sample Population (S=1)
Model 5: Direct Adjustment for Selection Covariate (L)	Logit(Y=1)=β_0 + β_1(E) + B_2(L)	-	Selected Sample Population (S=1)
Model 6: Unexposed Prognostic Weights using L and U (Stabilized)	Logit(Y=1)=β_0 + β_1(E)	$\frac{E[Y=y]}{E[Y=y\|L=l,U=u,E=0]}$	Selected Sample Population (S=1)
Model 7: Full Sample Prognostic Weights using L and U (Stabilized)	Logit(Y=1)=β_0 + β_1(E)	$\frac{E[Y=y]}{E^{**}[Y=y\|L=l,U=u,E=e]}$	Selected Sample Population (S=1)
Model 8: Modified Prognostic Weights using L and U (Stabilized)	Logit(Y=1)=β_0 + β_1(E)	$\frac{E[Y=y]}{E[Y=y\|L=l,U=u]}$	Selected Sample Population (S=1)
Model 9: Stabilized IPSW using E, L, and U	Logit(Y=1)=β_0 + β_1(E)	$\frac{E[S=s]}{E[S=s\|L=l,U=u,E=e]}$	Selected Sample Population (S=1)
Model 10: Direct Adjustment for Selection Covariate (L) and Unmeasured Selection Covariate (U)	Logit(Y=1)=β_0 + β_1(E) + B_2(L) + B_3(U)	-	Selected Sample Population (S=1)
Model 11: Crude Model in Full Population	Logit(Y=1)=β_0 + β_1(E)	-	Full Population (S=1, S=0)

Let E**[•]= Pred{E[Y=y|•, E=0]}. (Exposure variable is set to unexposed for prediction of weights).

3.3 Results

Simulation 1

The distributions of the effect estimates for each model of Simulation 1 where the association is moderate (conditional OR=3) are displayed in the top plot of Figure 10 with reference lines marking the conditional and marginal odds ratios. The bottom plot of Figure 10 is the distribution of the log robust variance estimates for each model for Simulation 1 with the moderate effect. Figure 11 depicts the percent relative bias for each model compared to the conditional and marginal OR in the top left and right plots, respectively, and log MSE for each model compared to the conditional and marginal OR in the bottom left and right plots. Table 12 provides a summary of the results of Simulation 1 when the conditional odds ratio is set to 3.

The mean marginal odds ratio across the 1,000 iterations of the simulation was found to be 2.59 (logOR=0.952). Model 4 uses inverse probability of selection weights (IPSW) and appears to yield an estimate of the marginal odds ratio. This is also seen in Model 6, which is the regression of Y on E in the full population i.e. those who were selected and not selected into the study. This can be thought of as the unbiased or true study population effect. The median relative percent bias comparing Model 4 (IPSW) and Model 6 (full population) to the marginal log odds ratio is 0.1614% and 2.782%, respectively. It should be noted that data from both the selected and unselected population on covariates that affect selection into the study is often not available in order to develop IPSW or obtain the effect estimate in the true population. In cases of immigrative selection

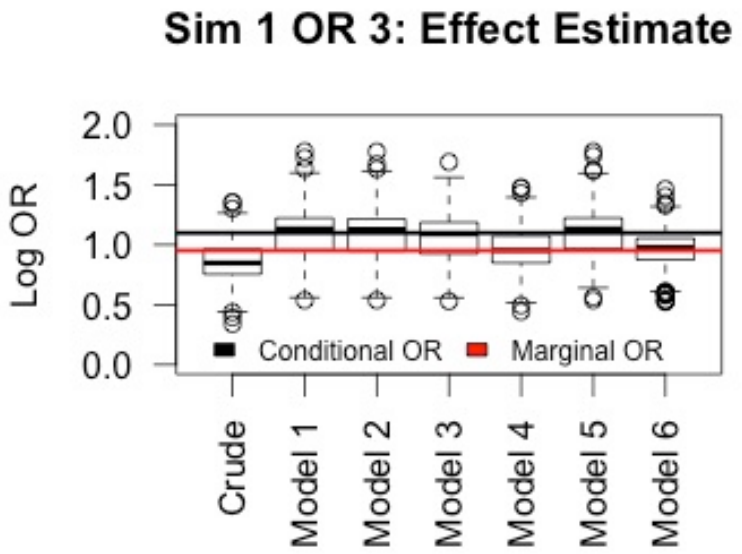
bias, Model 4 and Model 6 are unlikely to be used in analysis because of missing information about those selected into the study.

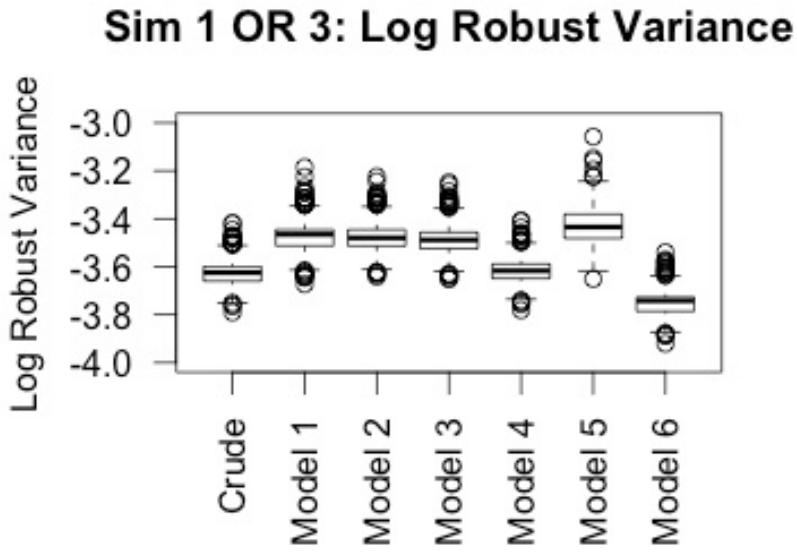
The models using the prognostic weights from the unexposed group, the full sample prognostic weights, the modified prognostic weights, and direct adjustment on L (Models 1-3 and 5) yield effect estimates closer to the conditional odds ratio. The direct adjustment model (Model 5) had the largest percent relative bias (3.095%) compared to the conditional OR, while the model using modified prognostic score weights had the lowest percent relative bias (-0.724%) compared to the conditional OR. When comparing these models to the marginal OR, the percent relative bias ranged from approximately 14.5 to 19%. It is interesting to note that the modified prognostic weights yield the marginal odds ratio in Chapter 2, when only confounders were present in the model; however, in Simulation 1, which includes only selection bias, the use of modified prognostic weights appears to yield in the conditional effect estimate.

The Monte Carlo variance estimates and median robust variance estimates are similar for all of the models. Apart from the crude model, Models 4 and 6 have the lowest variances. Model 5, which directly adjusts on L, has the highest variance using robust variance, while the unexposed prognostic weights (Model 1) has the highest variance using Monte Carlo variance.

Figure 10: Simulation 1 (Conditional OR=3) Effect Estimate and Log Robust Variance

Crude: Crude Model in Sample Population
Model 1: Unexposed Prognostic Score Weights (Stabilized)
Model 2: Full Sample Prognostic Score Weights (Stabilized)
Model 3: Modified Prognostic Score Weights (Stabilized)
Model 4: Stabilized IPSW (Inverse Probability of Selection Weights)
Model 5: Direct Adjustment on Selection Covariate (L)
Model 6: Crude Model in Full Population





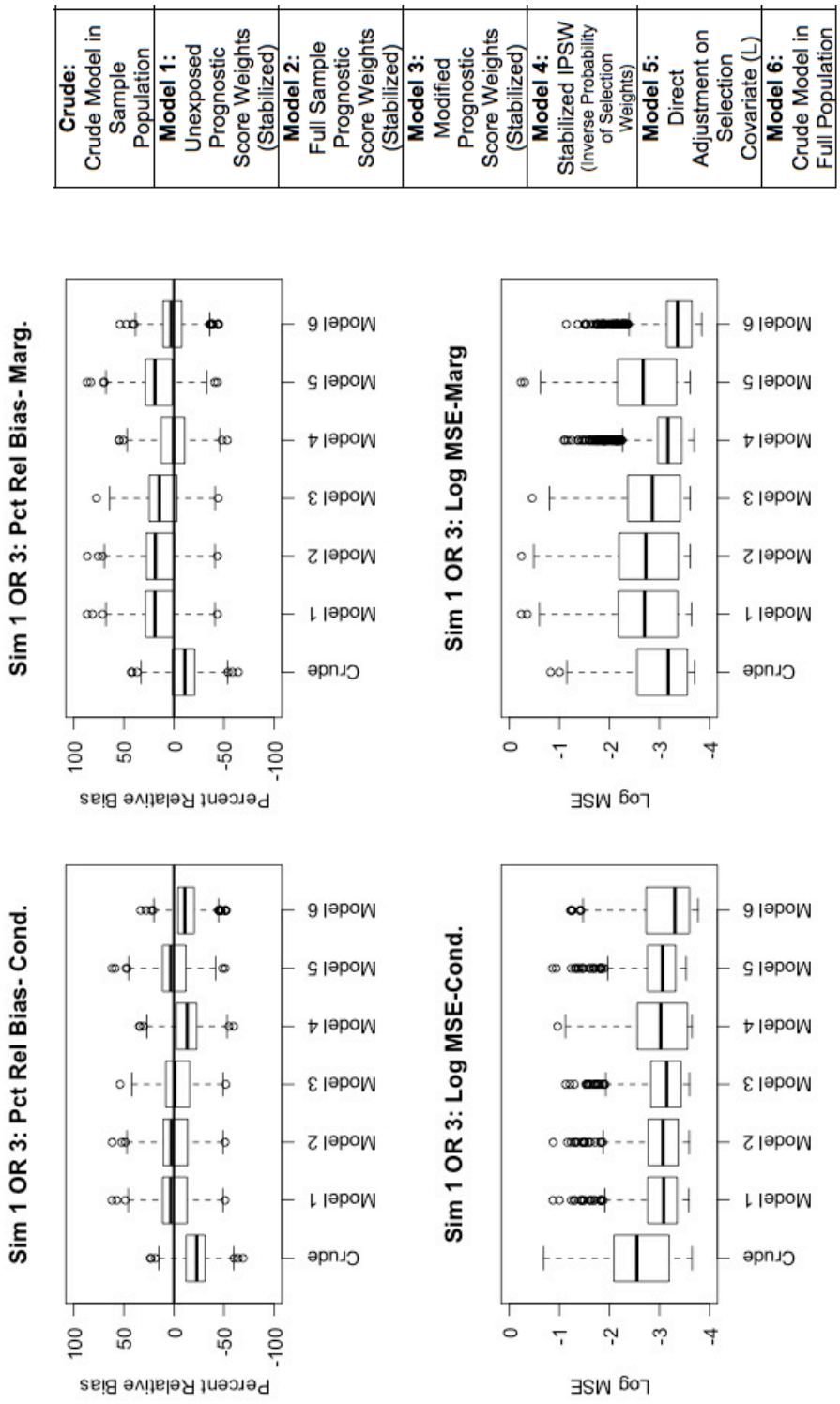


Figure 11: Simulation 1 (OR=3) Percent Relative Bias and Log Mean Squared Error

Table 12: Results for Simulation 1, OR=3

Description	Median logOR [95% CI Bootstrap]	Median Robust Variance	Median % Relative Bias (Conditional)	Median % Relative Bias (Marginal)	Median MSE (Conditional)	Median MSE (Marginal)	Monte Carlo Variance (bootstrap)
Crude: Crude Model in Sample Population	0.8477 [0.57, 1.14]	0.02666	-22.84	-10.911	0.07820	0.04171	0.02015
Model 1: Unexposed Prognostic Score Weights (Stabilized)	1.1322 [0.80, 1.40]	0.03127	3.060	18.9120	0.04573	0.06703	0.02968
Model 2: Full Sample Prognostic Score Weights (Stabilized)	1.1281 [0.81, 1.41]	0.03082	2.6864	18.482	0.04646	0.06531	0.02803
Model 3: Modified Prognostic Score Weights (Stabilized)	1.0907 [0.79, 1.37]	0.03055	-0.724	14.546	0.04310	0.05727	0.02458
Model 4: Stabilized IPSW (Inverse Probability of Selection Weights)	0.9534 [0.67, 1.26]	0.02687	-13.214	0.1614	0.04847	0.04199	0.02132
Model 5: Direct Adjustment on Selection Covariate (L)	1.1326 [0.80, 1.40]	0.03224	3.095	18.95	0.04688	0.06880	0.02846
Model 6: Crude Model in Full Population	0.9786 [0.69, 1.22]	0.02370	-10.920	2.782	0.03651	0.03464	0.01828

When Simulation 1 has a null effect (conditional OR=1), Figure 12 is the distribution of the effect estimates for each model and Figure 13 displays the log robust variance estimates and log MSE for each model. Table 13 is a summary of the results for Simulation 1 when the effect estimate is null.

In a model with a null effect estimate the marginal and conditional odds ratios are equivalent and thus the differences in conditional versus marginal effect estimates among the models for Simulation 1 when the OR was 3 are not seen. Models 1 through 6 approximate the conditional OR and the marginal odds ratio that was calculated in the simulation (both OR=1). The median effect estimate using the crude model is -0.154 while the median effect estimates of the other models range from -0.046 to -0.011.

Apart from Model 6, which had the true population data and by default a larger sample size, the Monte Carlo variance estimates and median robust variance estimates are similar across all of the models. It is of note that the unexposed prognostic score weights (Model 1) had the lowest Monte Carlo variance, lowest median robust variance, and lowest MSE among Models 1-6.

Figure 12: Simulation 1 (Conditional OR=1) Effect Estimate

Crude: Crude Model in Sample Population
Model 1: Unexposed Prognostic Score Weights (Stabilized)
Model 2: Full Sample Prognostic Score Weights (Stabilized)
Model 3: Modified Prognostic Score Weights (Stabilized)
Model 4: Stabilized IPSW (Inverse Probability of Selection Weights)
Model 5: Direct Adjustment on Selection Covariate (L)
Model 6: Crude Model in Full Population

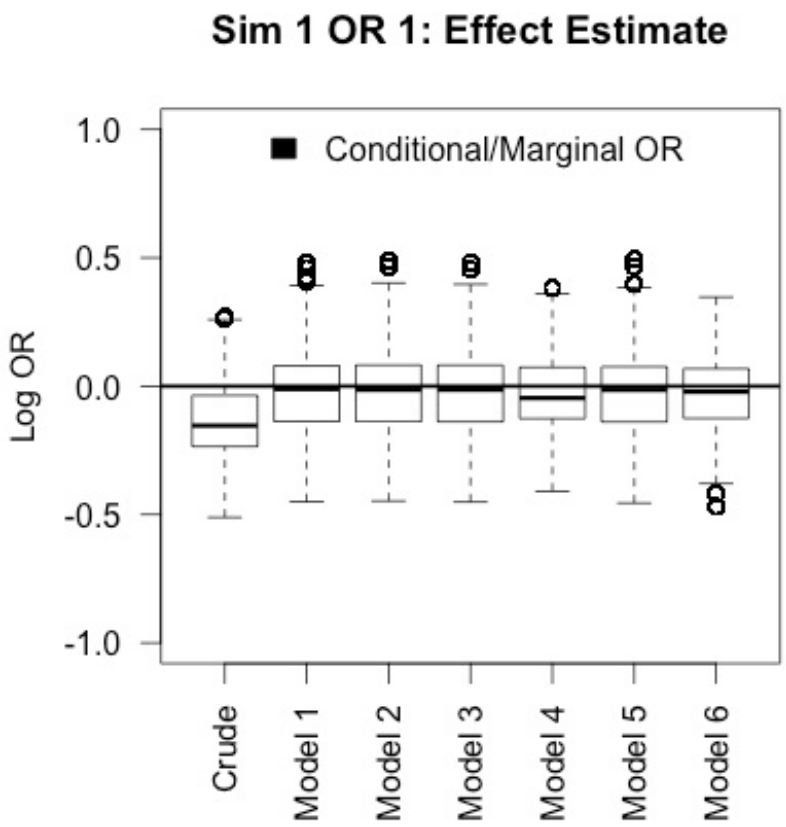
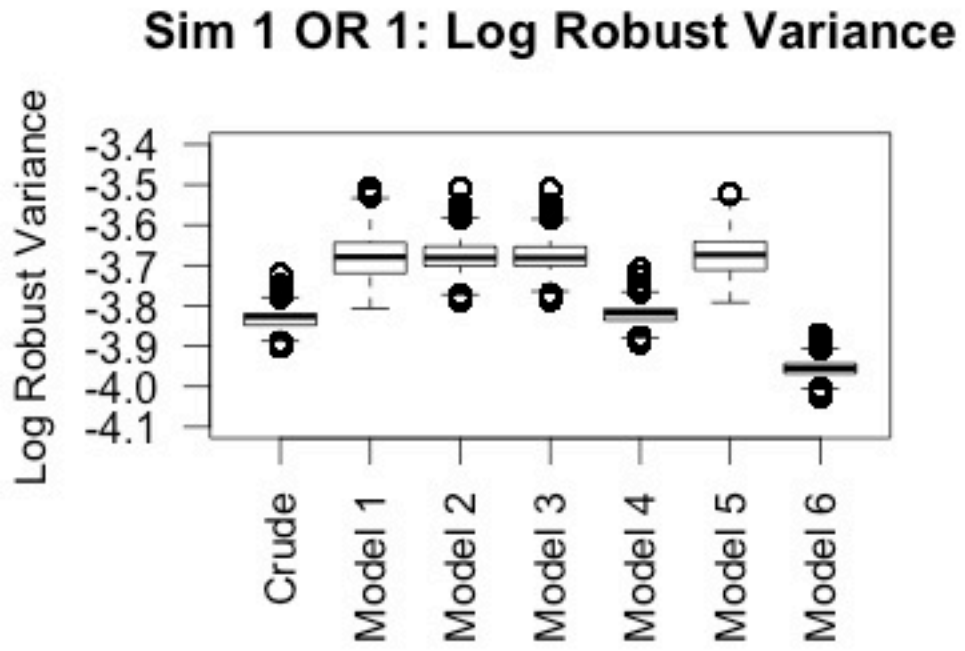


Figure 13: Simulation 1 (Conditional OR=1)
Log Robust Variance and Log Mean Squared Error



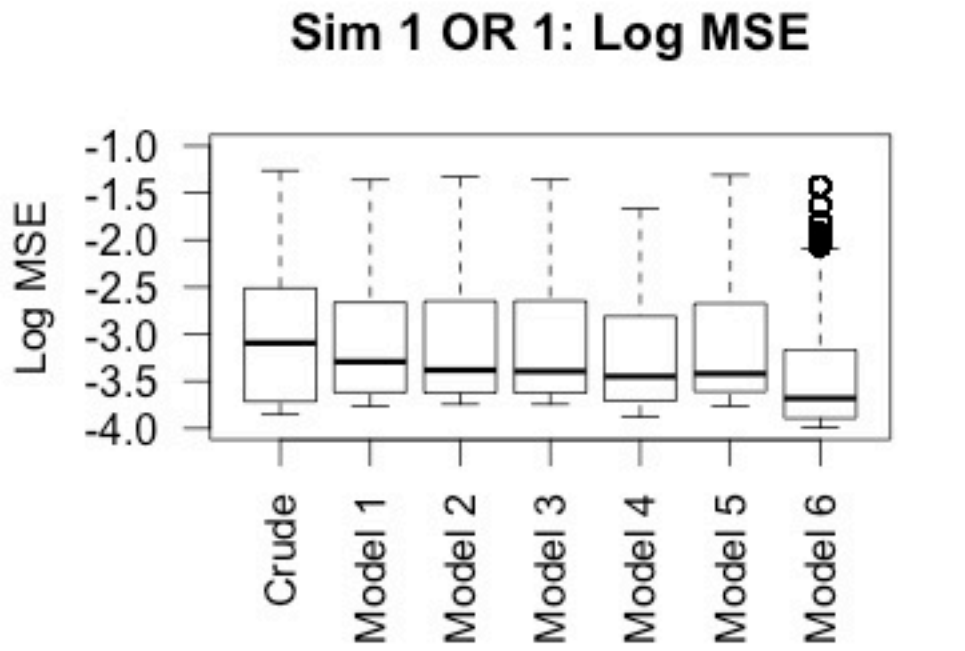


Table 13: Results for Simulation 1, OR=1

Description	Median logOR [95% CI Bootstrap]	Median Robust Variance	Median MSE (Conditional)	Monte Carlo Variance (bootstrap)
Crude: Crude Model in Sample Population	-0.15362 [-0.50, 0.15]	0.02178	0.04538	0.027
Model 1: Unexposed Prognostic Score Weights (Stabilized)	-0.01114 [-0.45, 0.28]	0.02524	0.03720	0.034066
Model 2: Full Sample Prognostic Score Weights (Stabilized)	-0.01407 [-0.45, 0.28]	0.02520	0.03399	0.03323
Model 3: Modified Prognostic Score Weights (Stabilized)	-0.01391 [-0.45, 0.27]	0.02519	0.03358	0.03336
Model 4: Stabilized IPSW (Inverse Probability of Selection Weights)	-0.04641 [-0.39, 0.25]	0.02197	0.03179	0.02808
Model 5: Direct Adjustment on Selection Covariate (L)	-0.01448 [-0.46, 0.27]	0.02538	0.03272	0.03338
Model 6: Crude Model in Full Population	-0.02170 [-0.27, 0.22]	0.01916	0.02521	0.01666

Simulation 2

The results for each of the models in Simulation 2 where the effect is moderate (conditional OR=3) are displayed in Figures 14-16. Figure 14 is a boxplot of the effect estimates with reference lines marking the conditional and marginal odds ratios. Figure 15 is a boxplot of the log robust variance estimates. Figure 16 are boxplots of the percent relative bias (top) and log MSE (bottom) for the models that control for both confounding and selection bias (Models 1-7) compared to the conditional (left) and marginal OR (right). Models 9, 10, and 11 are presented to provide information based on the bias that can be attributed to selection bias when confounding is controlled for and vice versa. Model 8 yields the effect estimate based on direct adjustment for the confounder among the selected and unselected observations. Table 14 provides a summary of the results of Simulation 2 when the conditional odds ratio is set to 3.

The mean marginal odds ratio across the 1,000 iterations of the simulation was 2.571 (logOR=0.944). The unexposed prognostic score based on L and C (Model 1), full sample prognostic score based on L and C (Model 2), combination weights using the unexposed prognostic score and IPEW (Model 4), combination weights using the full population prognostic score and IPEW (Model 5), and direct adjustment (Model 7) appear yield the conditional odds ratio based on the figures. The median percent relative bias from Table 14 for Models 1, 2, 4, 5, and 7 all have an absolute value of less than 1.5% when compared to the conditional OR. The modified prognostic score (Model 3) appears to yield a result somewhere in between the marginal and conditional OR with a median percent

relative bias of -7.99% compared to the conditional OR and 7.08% compared to the marginal OR. There is also an increased bias observed when using the combination weights of the modified prognostic score and IPEW (Model 6) with a median percent relative bias of -2.85% compared to the conditional OR and 13.25% compared to the marginal OR. The observed bias is likely reduced by the inclusion of IPEW weights when compared to only the modified prognostic model (Model 3).

None of the models that control for selection bias and confounding (Models 1-7) appear to yield the marginal odds ratio. It is likely that combined weights using IPEW and IPSW could yield the marginal for this DAG. The Monte Carlo variance estimates and median robust variance estimates are similar for all of the models that address both selection and confounding with values ranging from 0.022 to 0.03. Interestingly the models using the modified prognostic score (Models 3 and 6), which did not appear to estimate either the conditional or the marginal OR, had the lowest variance estimates. The model that directly adjusts for L and C (Model 7) has the highest median robust variance estimate, while the unexposed prognostic score (Model 1) has the highest Monte Carlo variance estimate. The full sample prognostic score model (Model 2) had the lowest MSE.

Figure 14: Simulation 2 (Conditional OR=3) Effect Estimate

Crude: Crude Model
Model 1: Unexposed Prog.Weights using L and C (Stabilized)
Model 2: Full Sample Prog. Weights using L and C (Stabilized)
Model 3: Modified Prog. Weights using L and C (Stabilized)
Model 4: Combination Weights (Stabilized): IPEW using C; Unexposed Prog. Weights using L
Model 5: Combination Weights (Stabilized): IPEW using C; Full Sample Prog. Weights using L
Model 6: Combination Weights (Stabilized): IPW using C; Modified Prog. Weights using L
Model 7: Direct Adjustment for L and C
Model 8: Direct Adjustment for C among Full Population (No Selection)
Model 9: Crude Model among Full Population (Confounded)
Model 10: Direct Adjustment for C among Sample Population (Selection Bias)
Model 11: Direct Adjustment for L among Sample Population (Confounded)

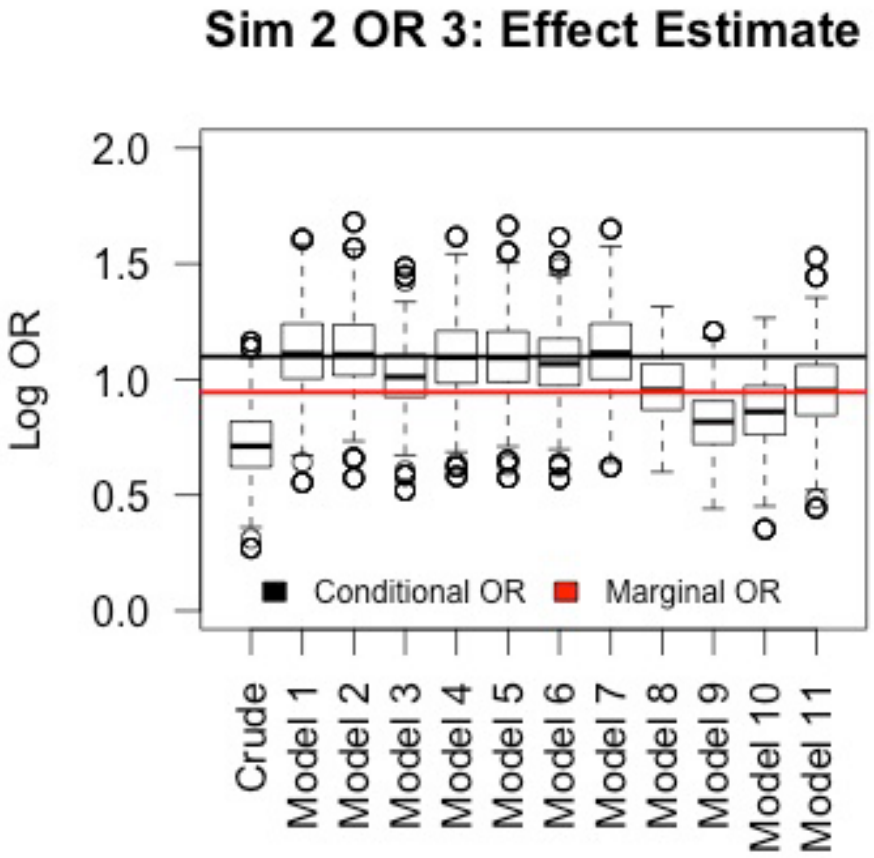
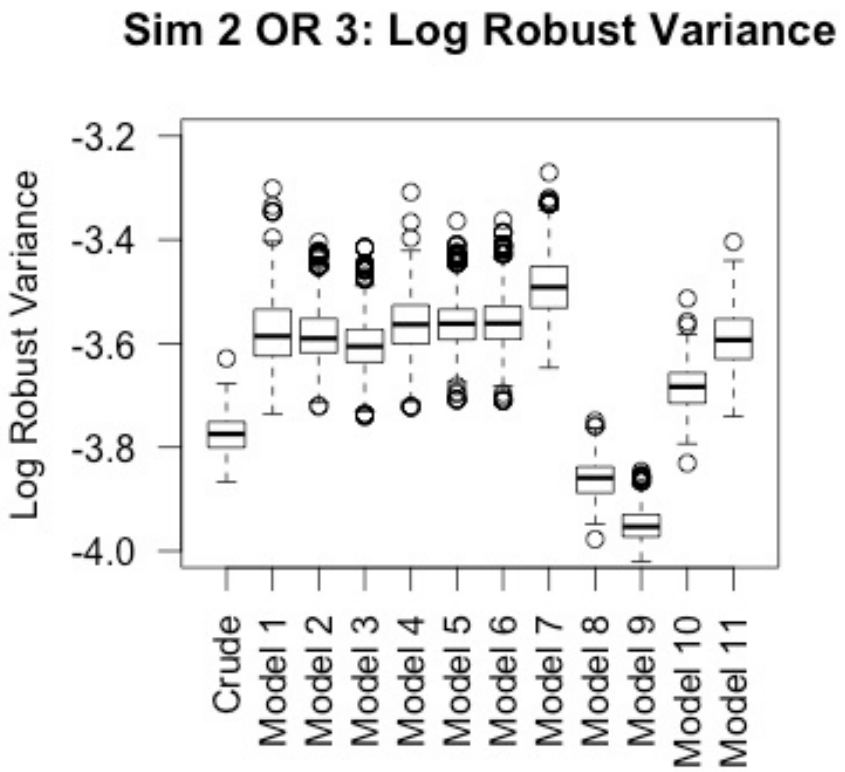


Figure 15: Simulation 2 (Conditional OR=3) Log Robust Variance

Crude: Crude Model
Model 1: Unexposed Prog.Weights using L and C (Stabilized)
Model 2: Full Sample Prog. Weights using L and C (Stabilized)
Model 3: Modified Prog. Weights using L and C (Stabilized)
Model 4: Combination Weights (Stabilized): IPEW using C; Unexposed Prog. Weights using L
Model 5: Combination Weights (Stabilized): IPEW using C; Full Sample Prog. Weights using L
Model 6: Combination Weights (Stabilized): IPW using C; Modified Prog. Weights using L
Model 7: Direct Adjustment for L and C
Model 8: Direct Adjustment for C among Full Population (No Selection)
Model 9: Crude Model among Full Population (Confounded)
Model 10: Direct Adjustment for C among Sample Population (Selection Bias)
Model 11: Direct Adjustment for L among Sample Population (Confounded)



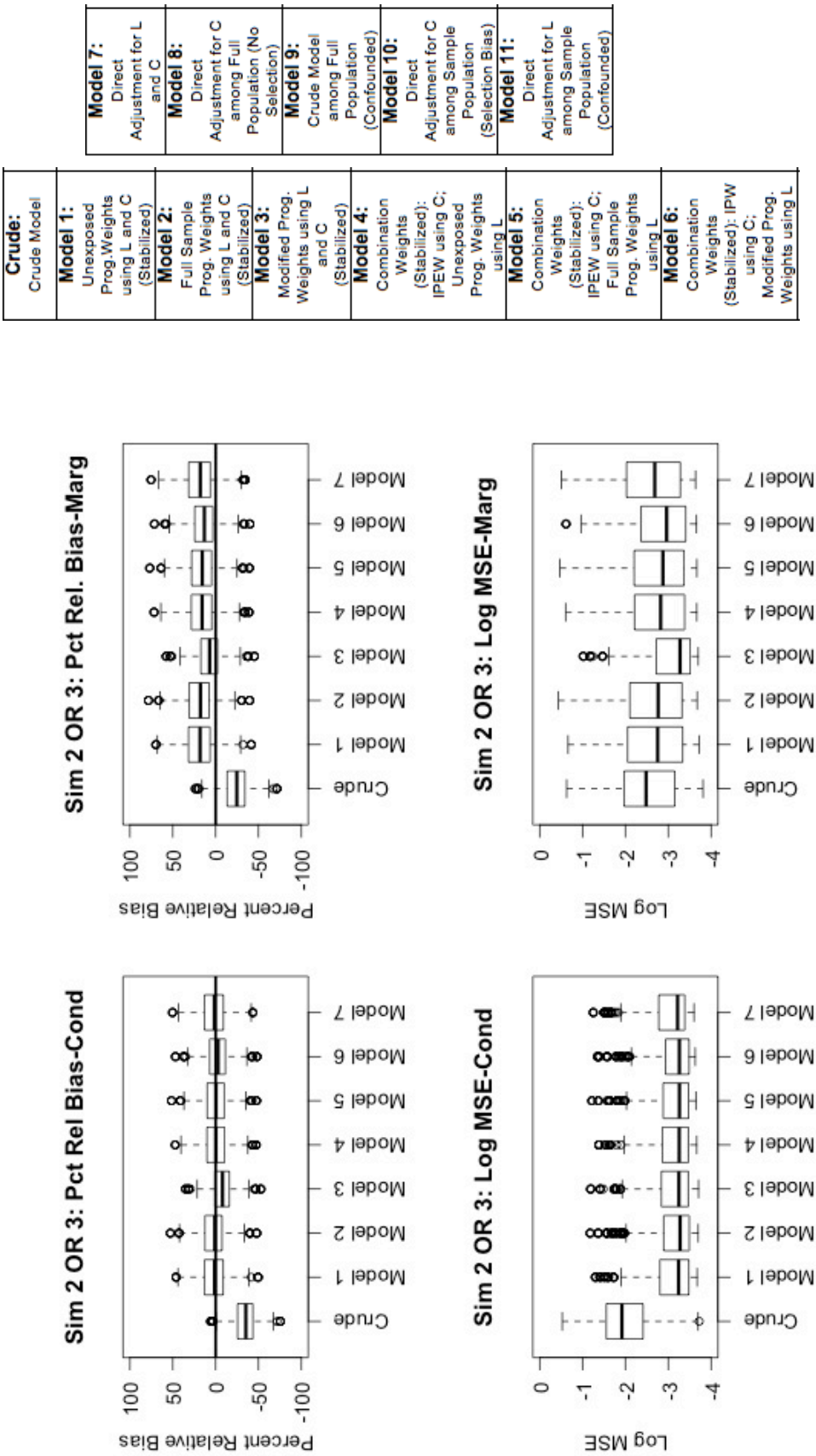


Figure 16: Simulation 2 (OR=3) Percent Relative Bias and Log Mean Squared Error

Table 14: Results for Simulation 2, OR=3

Description	Median logOR [95% CI Bootstrap]	Median Robust Variance	Median % Relative Bias (Conditional)	Median % Relative Bias (Marginal)	Median MSE (Conditional)	Median MSE (Marginal)	Monte Carlo Variance (bootstrap)
Crude: Crude Model	0.7119 [0.41, 0.98]	0.02295	-35.202	-24.68	0.14670	0.08352	0.02225
Model 1: Unexposed Prog.Weights using L and C (Stabilized)	1.1117 [0.75, 1.45]	0.02771	1.193	18.175	0.03907	0.06367	0.0297
Model 2: Full Sample Prog. Weights using L and C (Stabilized)	1.1089 [0.78, 1.49]	0.02759	0.9393	17.798	0.03776	0.06299	0.02897
Model 3: Modified Prog. Weights using L and C (Stabilized)	1.0108 [0.72, 1.31]	0.02717	-7.9911	7.081	0.03879	0.03780	0.0222
Model 4: Combination Weights (Stabilized): IPEW using C; Unexposed Prog. Weights using L	1.0955 [0.76, 1.40]	0.02834	-0.2807	15.872	0.03852	0.05939	0.02771
Model 5: Combination Weights (Stabilized): IPEW using C; Full Sample Prog. Weights using L	1.0941 [0.77, 1.43]	0.02839	-0.41318	15.659	0.03834	0.05638	0.02754
Model 6: Combination Weights (Stabilized): IPW using C; Modified Prog. Weights using L	1.0673 [0.76, 1.40]	0.02841	-2.854	13.148	0.03839	0.05166	0.02531
Model 7: Direct Adjustment for L and C	1.1149 [0.78, 1.44]	0.03046	1.485	17.936	0.040	0.06813	0.02864
Model 8: Direct Adjustment for C among Full Population (No Selection)	0.9546 [0.72, 1.22]	0.02109	-	-	-	-	0.01829
Model 9: Crude Model among Full Population (Confounded)	0.8171 [0.57, 1.12]	0.01920	-	-	-	-	0.0182
Model 10: Direct Adjustment for C among Sample Population (Selection Bias)	0.8608 [0.57, 1.16]	0.02514	-	-	-	-	0.02235
Model 11: Direct Adjustment for L among Sample Population (Confounded)	0.9525 [0.62, 1.26]	0.02752	-	-	-	-	0.02798

For the null effect (conditional OR=1) in Simulation 2, Figure 16 is the distribution of the effect estimates for all the models, Figure 17 is the log robust variance estimates for all the models, and Figure 38 is the log MSE compared to the conditional OR for the models that address both selection bias and confounding (Models 1-7). The calculations and figures were created using log odds ratios for each model in Simulation 2. Models 9, 10, and 11 estimate the bias that can be attributed solely to either selection or confounding bias. Model 8 yields the effect estimate based on direct adjustment for the confounder among the selected and unselected observations. Table 15 is a summary of the results for Simulation 2 when the effect estimate is null (OR=1).

The observed differences between the models for Simulation 2 based on the marginal and conditional OR are not seen with a null effect estimate because the marginal and conditional OR are equivalent. Models 1 through 7 approximate the marginal odds ratio and all have approximately a 1% or lower median percent relative bias compared to a -23.1% relative bias in the crude model. The Monte Carlo variance estimates and median robust variance estimates are similar for all of the models that address both selection bias and confounding (Models 1-7). As observed in the first part of Simulation 2, the model that directly adjusts for L and C (Model 7) has the highest median robust variance estimate, while the unexposed prognostic score (Model 1) has the highest Monte Carlo variance estimate. Again, the full sample prognostic score model (Model 2) was observed to have the lowest MSE.

Figure 17: Simulation 2 (Conditional OR=1) Effect Estimate

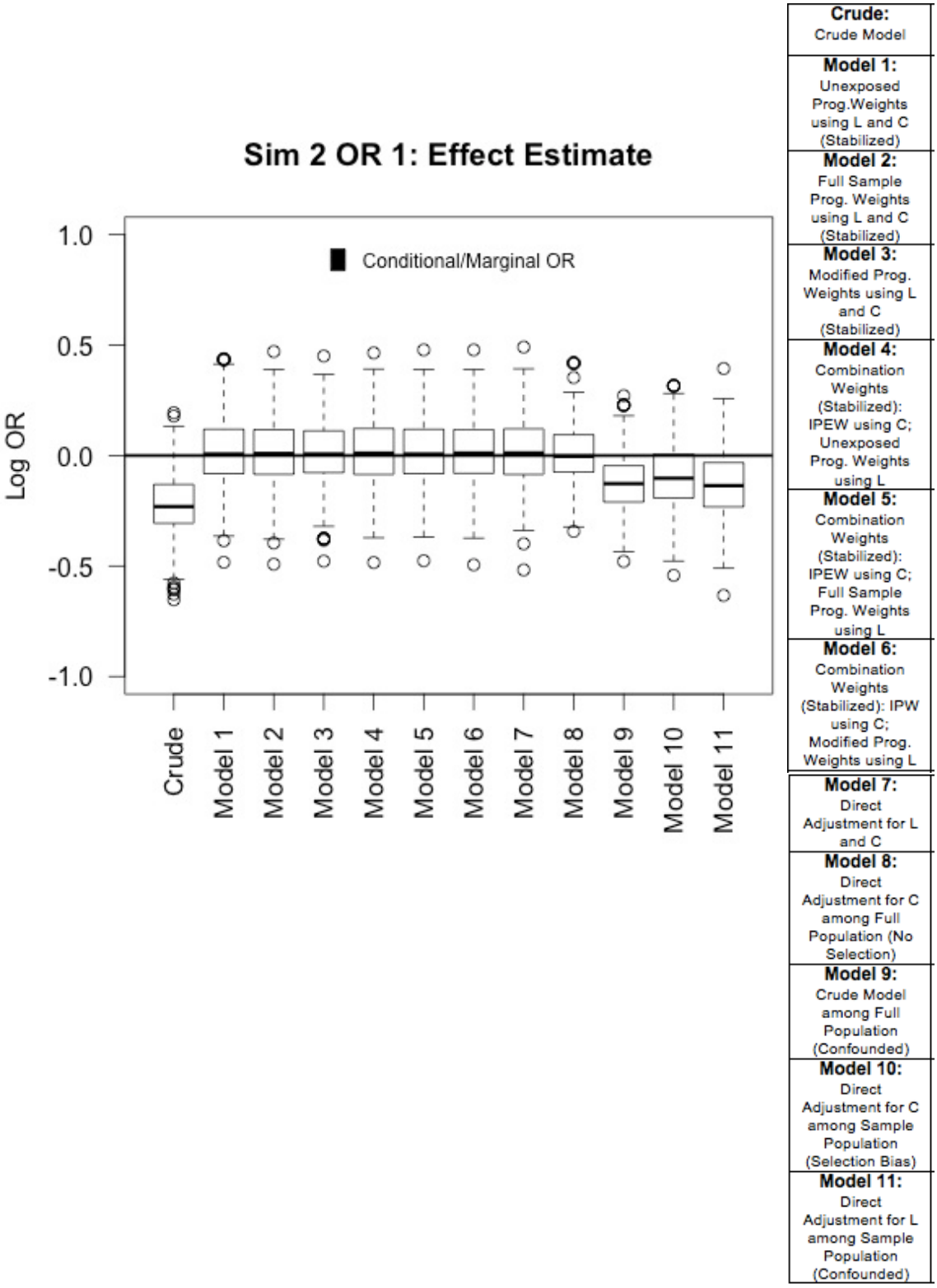


Figure 18: Simulation 2 (Conditional OR=1) Log Robust Variance

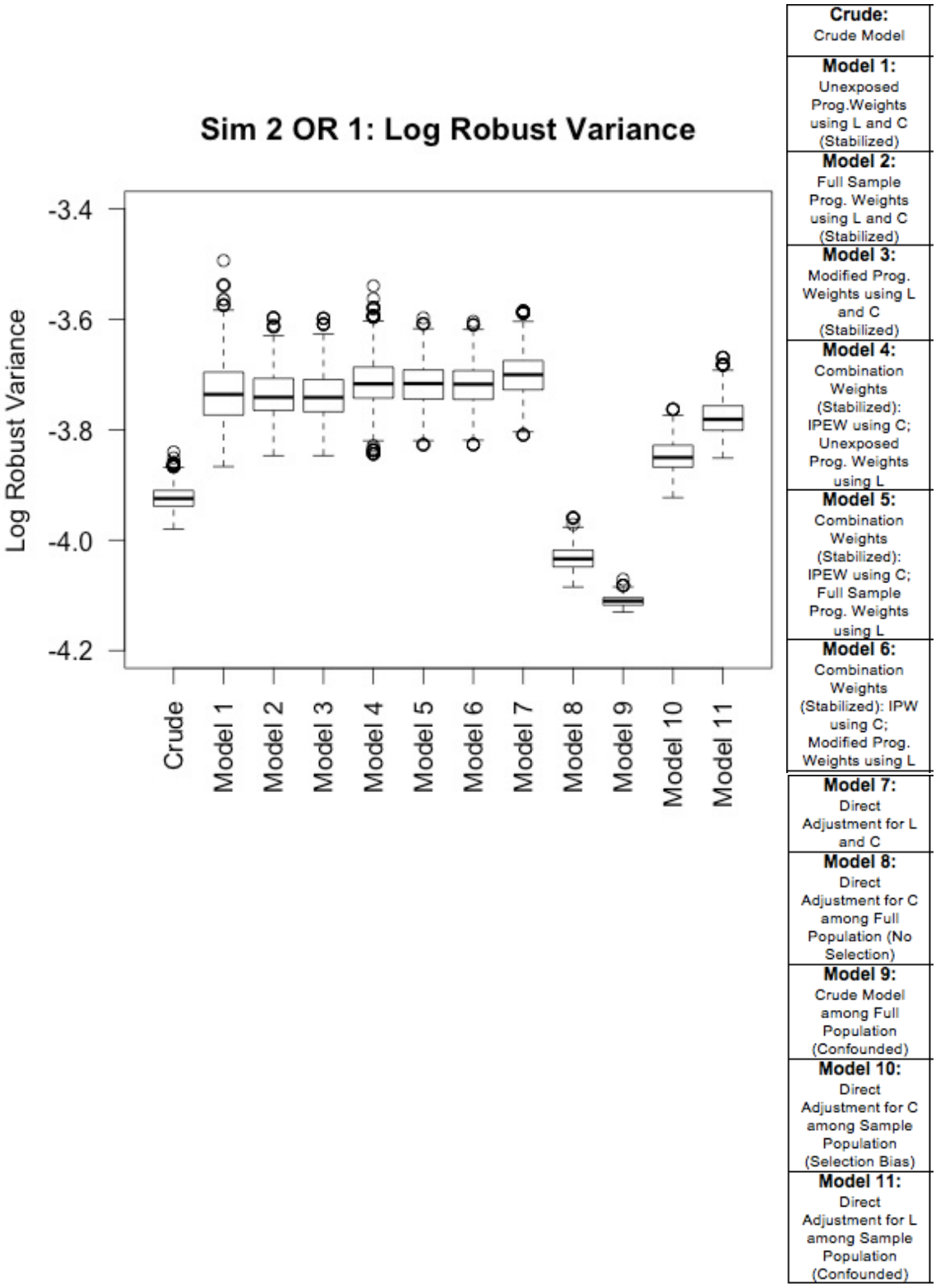


Figure 19: Simulation 2 (Conditional OR=1) Log Mean Squared Error

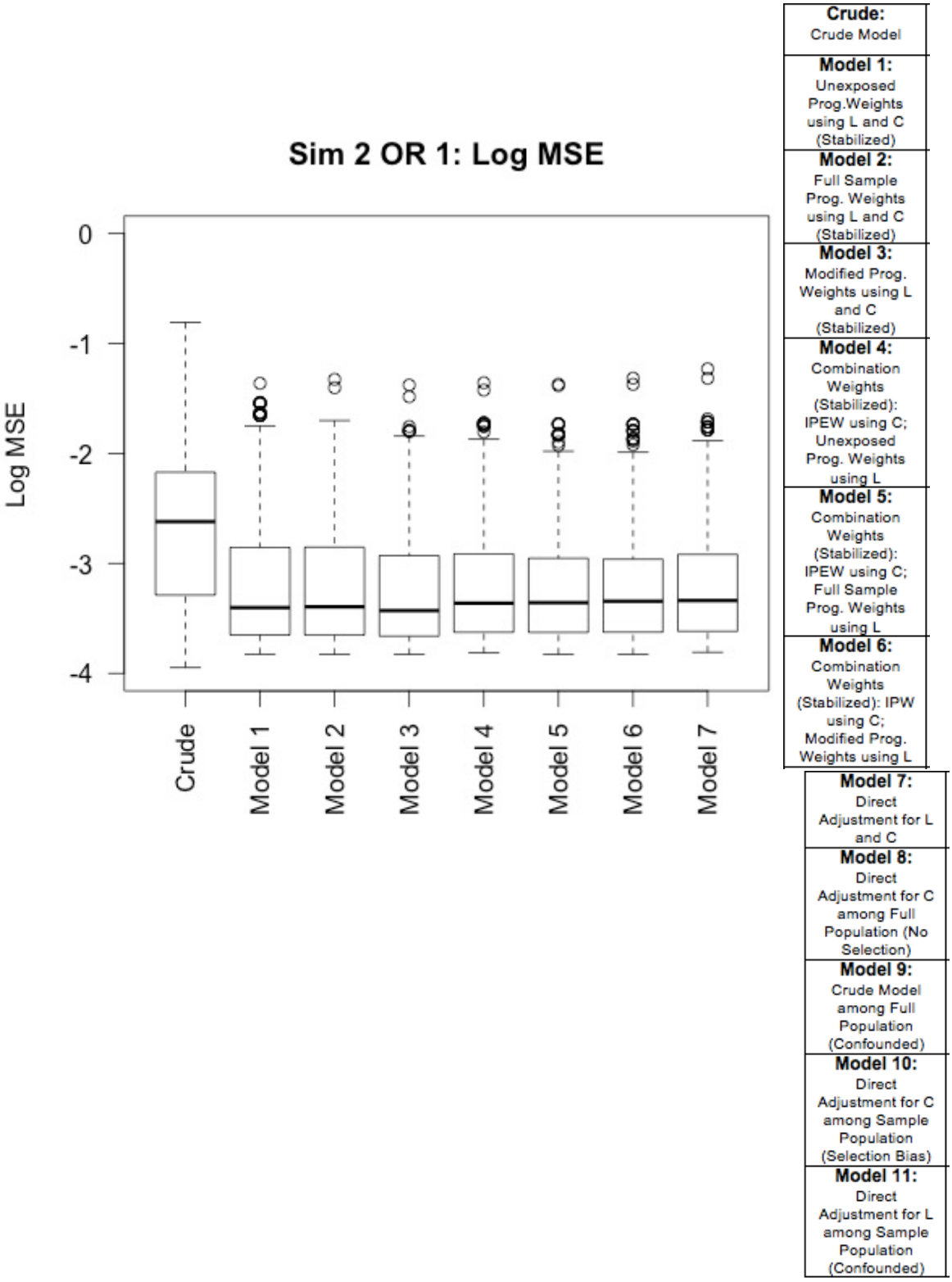


Table 15: Results for Simulation 2, OR=1

Description	Median logOR [95% CI Bootstrap]	Median Robust Variance	Median MSE (Conditional)	Monte Carlo Variance (bootstrap)
Crude: Crude Model	-0.2310 [-0.51, 0.04]	0.01975	0.07277	0.01806
Model 1: Unexposed Prog.Weights using L and C (Stabilized)	0.00699 [-0.29, 0.30]	0.02385	0.03329	0.0229
Model 2: Full Sample Prog. Weights using L and C (Stabilized)	0.00890 [-0.29, 0.34]	0.02373	0.03353	0.0226
Model 3: Modified Prog. Weights using L and C (Stabilized)	0.00702 [-0.27, 0.32]	0.02372	0.03240	0.01969
Model 4: Combination Weights (Stabilized): IPEW using C; Unexposed Prog. Weights using L	0.01062 [-0.27, 0.29]	0.02431	0.03466	0.02094
Model 5: Combination Weights (Stabilized): IPEW using C; Full Sample Prog. Weights using L	0.006887 [-0.27, 0.32]	0.02432	0.03482	0.020921
Model 6: Combination Weights (Stabilized): IPW using C; Modified Prog. Weights using L	0.01097 [-0.27, 0.32]	0.02430	0.03529	0.020763
Model 7: Direct Adjustment for L and C	0.01147 [-0.27, 0.33]	0.02472	0.03550	0.02181
Model 8: Direct Adjustment for C among Full Population (No Selection)	-0.003305 [-0.24, 0.24]	0.01771	-	0.01543
Model 9: Crude Model among Full Population (Confounded)	-0.12717 [-0.37, 0.11]	0.01640	-	0.01486
Model 10: Direct Adjustment for C among Sample Population (Selection Bias)	-0.102730 [-0.38, 0.15]	0.02127	-	0.01845
Model 11: Direct Adjustment for L among Sample Population (Confounded)	-0.13617 [-0.42, 0.13]	0.02280	-	0.02075

Simulation 3

The results for each of the models in Simulation 3 where the effect is moderate (conditional OR=3) are displayed in Figures 18-21. Figure 18 is the effect estimates and Figure 19 is the log robust variance estimates. Figures 20 shows the percent relative bias compared to the conditional (left) and marginal (right) ORs for the prognostic score models based on L (Models 1-5) in the top panels and for the prognostic score models based on U and L in the bottom panels. Figure 21 shows the log MSE compared to the conditional (left) and marginal (right) ORs for the prognostic score models based on L (Models 1-5) in the top panels and for the prognostic score models based on U and L in the bottom panels. Table 16 summarizes the results of Simulation 3 when the odds ratio is set to 3.

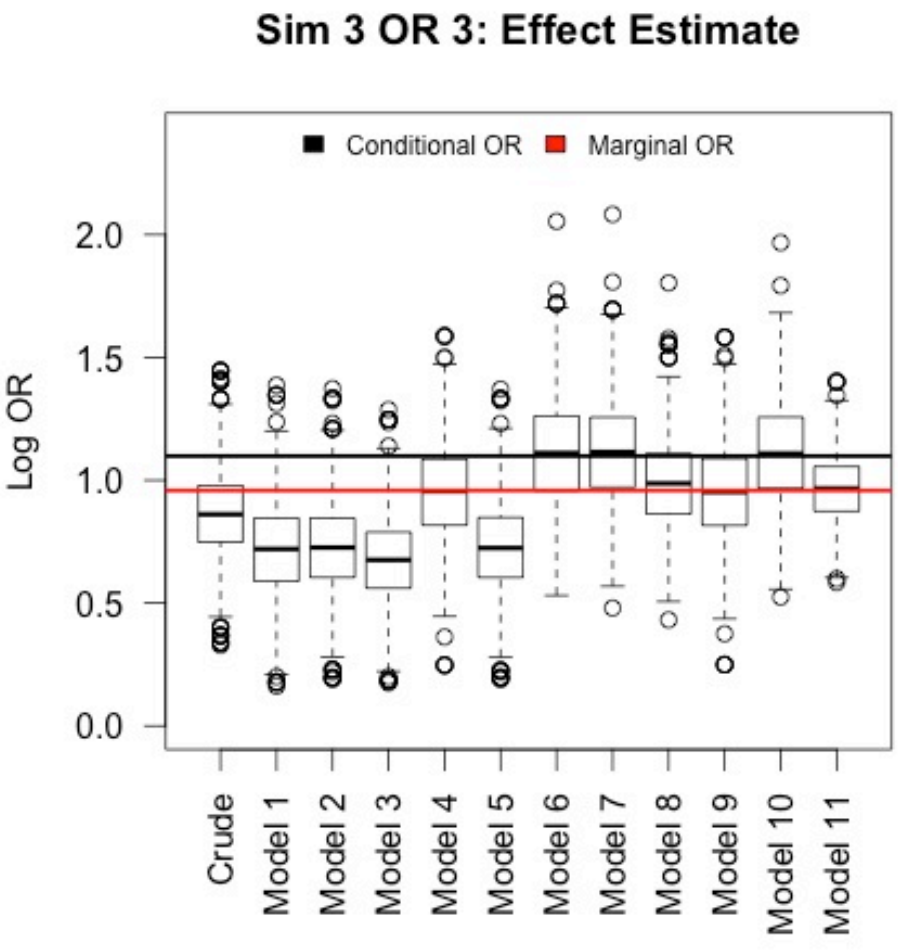
The mean marginal odds ratio across the 1,000 iterations of the simulation was 2.608 (logOR=0.959). The model results in this simulation can be separated by whether or not they use the covariate U in the model. U represents an unmeasured selection variable, so in practice only the crude model and Models 1-5 could be included in the analysis. It should also be noted that Model 4 and Model 9, based on IPSW, require that information be known regarding those who are not selected into the study. This is often a limitation in situations with immigrative selection bias. Of Models 1-5, only Model 4 that uses IPSW yields an unbiased estimate. Model 4 approximates the marginal odds ratio with a median percent relative bias on -0.39%. The three prognostic models based only on L (Models 1, 2, and 3) and Model 5, which directly adjusts on L, result in more bias

than the crude model. The median percent relative bias for these models is higher than 30% when compared to the conditional OR and higher than approximately 25% when compared to the marginal OR.

Models 6 through 10 includes estimates using the three prognostic scores weights based on U and L, IPSW using U and L, and direct adjustment for U and L. If U had been measured, then Models 6 through 10 could have been included in the analysis resulting in similar conclusions to Simulation 1 and 2. Model 11 is the crude regression model performed in the full population and should represent the effect estimate in the absence of selection bias but with confounding bias. For Models 6-10, the unexposed prognostic score weights, full population prognostic score weights, and direct adjustment model appear to estimate the conditional OR, while IPSW estimates the marginal OR. As seen in Simulation 2, the modified prognostic score has a higher percent relative bias than the other models when compared to the marginal and conditional OR. It is likely that it yields and effect estimate somewhere in between the marginal and conditional OR.

The Monte Carlo variance estimates and median robust variance estimates are similar for all of the models regardless of whether or not they include U. As in Simulation 2, the model that directly adjusts for L and U (Model 10) has the highest median robust variance estimate, while the unexposed prognostic score using L and U (Model 6) has the highest Monte Carlo variance. Models 6-10 (based on U and L) performed better by MSE than Models 1-5 (based on L only), which is intuitive since they were more biased than the crude.

Figure 20: Simulation 3 (Conditional OR=3)
Effect Estimate



Crude: Crude Model

Model 1: Unexposed Prognostic Weights using L (Stabilized)

Model 2: Full Sample Prognostic Weights using L (Stabilized)

Model 3: Modified Prognostic Weights using L (Stabilized)

Model 4: Stabilized IPSW using E and L

Model 5: Direct Adjustment for L

Model 6: Unexposed Prognostic Weights using L and U (Stabilized)

Model 7: Full Sample Prognostic Weights using L and U (Stabilized)

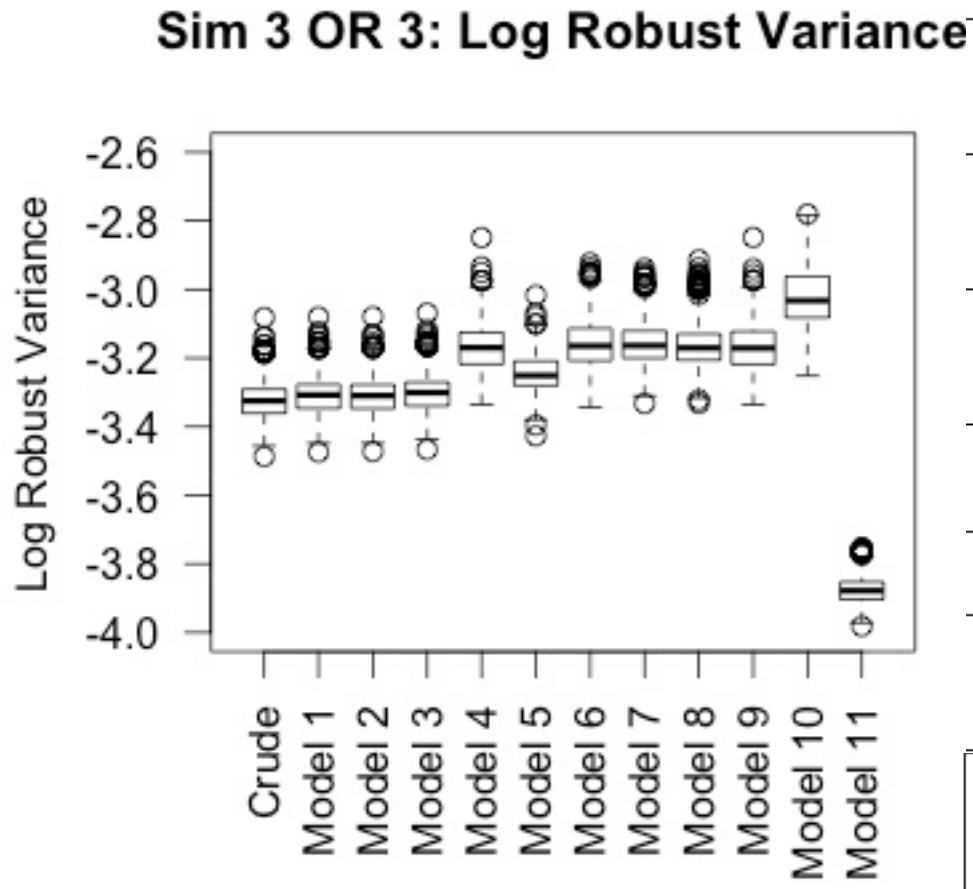
Model 8: Modified Prognostic Weights using L and U (Stabilized)

Model 9: Stabilized IPSW using E, L, and U

Model 10: Direct Adjustment for L and U

Model 11: Crude Model in Full Population

Figure 21: Simulation 3 (Conditional OR=3)
Log Robust Variance



Crude: Crude Model

Model 1: Unexposed Prognostic Weights using L (Stabilized)

Model 2: Full Sample Prognostic Weights using L (Stabilized)

Model 3: Modified Prognostic Weights using L (Stabilized)

Model 4: Stabilized IPSW using E and L

Model 5: Direct Adjustment for L

Model 6: Unexposed Prognostic Weights using L and U (Stabilized)

Model 7: Full Sample Prognostic Weights using L and U (Stabilized)

Model 8: Modified Prognostic Weights using L and U (Stabilized)

Model 9: Stabilized IPSW using E, L, and U

Model 10: Direct Adjustment for L and U

Model 11: Crude Model in Full Population

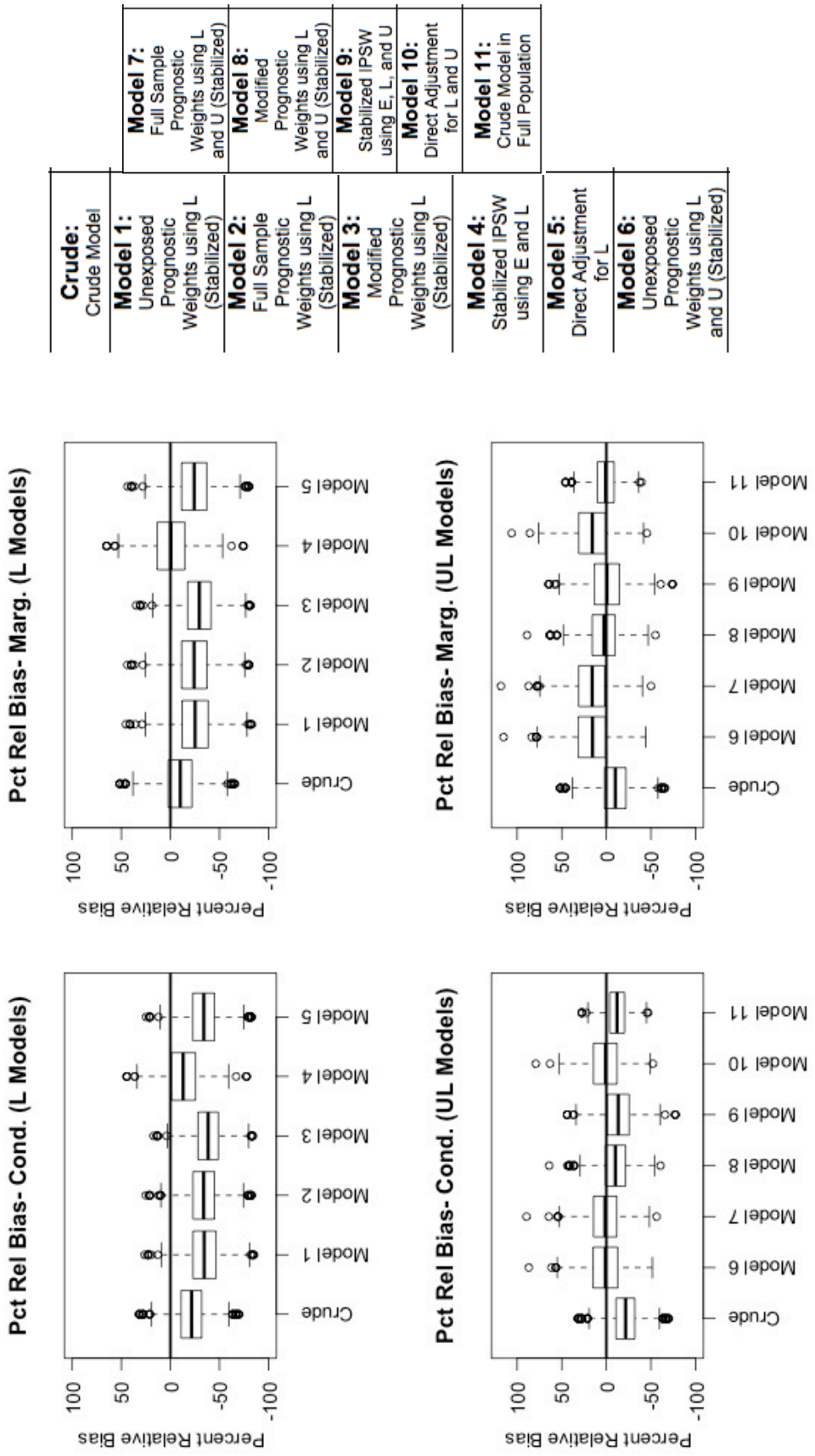


Figure 22: Simulation 3 (Conditional OR=3) Percent Relative Bias

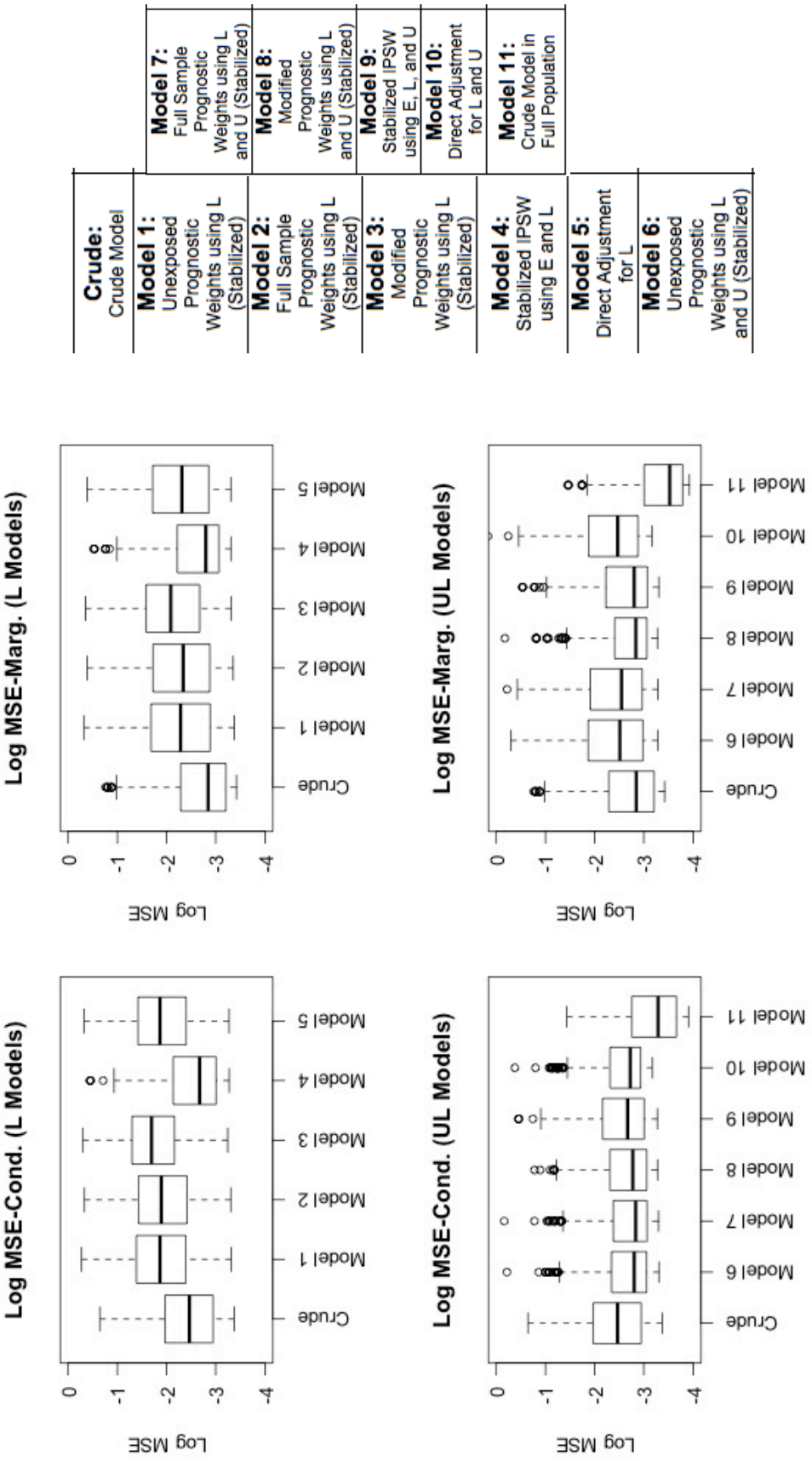


Figure 23: Simulation 3 (Conditional OR=3) Log Mean Squared Error

Table 16: Results for Simulation 3, OR=3

Description	Median logOR [95% CI Bootstrap]	Median Robust Variance	Median % Relative Bias (Conditional)	Median % Relative Bias (Marginal)	Median MSE (Conditional)	Median MSE (Marginal)	Monte Carlo Variance (bootstrap)
Crude: Crude Model	0.8614 [0.47, 1.26]	0.03598	-21.59	-10.085	0.08507	0.05817	0.0374
Model 1: Unexposed Prognostic Weights using L (Stabilized)	0.7202 [0.33, 1.11]	0.03660	-34.44	-25.17	0.15479	0.10184	0.03906
Model 2: Full Sample Prognostic Weights using L (Stabilized)	0.7260 [0.35, 1.13]	0.03653	-33.91	-24.14	0.15072	0.09656	0.03918
Model 3: Modified Prognostic Weights using L (Stabilized)	0.6757 [0.32, 1.07]	0.03685	-38.50	-29.52	0.18371	0.12427	0.03445
Model 4: Stabilized IPSW using E and L	0.9555 [0.58, 1.43]	0.04201	-13.022	-0.3861	0.06915	0.06092	0.04543
Model 5: Direct Adjustment for L	0.7242 [0.35, 1.14]	0.03874	-34.08	-24.25	0.1548	0.09916	0.03928
Model 6: Unexposed Prognostic Weights using L and U (Stabilized)	1.1112 [0.69, 1.56]	0.04223	1.142	15.4939	0.06063	0.08118	0.04968
Model 7: Full Sample Prognostic Weights using L and U (Stabilized)	1.1149 [0.68, 1.56]	0.04227	1.487	15.884	0.05879	0.07850	0.04836
Model 8: Modified Prognostic Weights using L and U (Stabilized)	0.9873 [0.59, 1.38]	0.04198	-10.132	2.701	0.06231	0.05865	0.03674
Model 9: Stabilized IPSW using E, L, and U	0.9523 [0.59, 1.43]	0.04196	-13.3213	-0.7042	0.06937	0.06065	0.04564
Model 10: Direct Adjustment for L and U	1.1089 [0.69, 1.55]	0.04819	0.9399	15.7560	0.06570	0.08476	0.04843
Model 11: Crude Model in Full Population	0.9669 [0.66, 1.24]	0.02069	-11.989	0.7866	0.03730	0.02944	0.02006

The results for Simulation 3 with the null effect (conditional OR=1) are displayed in Figure 22. The top left plot in the figure is the distribution of the effect estimates for all the models, the top right plot is the log robust variance estimates for all the models, the bottom left plot is the log MSE for the models based on L only (Models 1-5), and the bottom right plot is the log MSE for the models based on L and U (Models 1-5). Table 17 is a summary of the results for Simulation 3 when the effect estimate is null (OR=1).

Models 4 and 9, based on IPSW, require that information be known regarding those who are not selected into the study in order to estimate the probability of selection. As in the prior simulation, U represents an unmeasured variable and thus only models 1 through 5 could be performed in an analysis. Of these models, only Model 4 yields an unbiased estimate with a median percent relative bias of -1.076%. The three prognostic models based only on L (Models 1, 2, and 3) and Model 5, which directly adjusts on L, again result in more bias than the crude model. The log OR estimated from the crude model is -0.089, while the log ORs estimated using models 1, 2, 3, and 5 are approximately -0.2.

Models 6 through 11 includes estimates using the three prognostic scores weights based on U and L, direct adjustment for U and L, and IPSW using U and L. These models similar conclusions to Simulation 1 and 2 when the effect estimate is null. The Monte Carlo variance estimates and robust variance estimates are similar for all of the models. The direct adjustment model has the highest median robust variance estimate and the unexposed prognostic score using L and U (Model 6) has the highest Monte Carlo variance estimate.

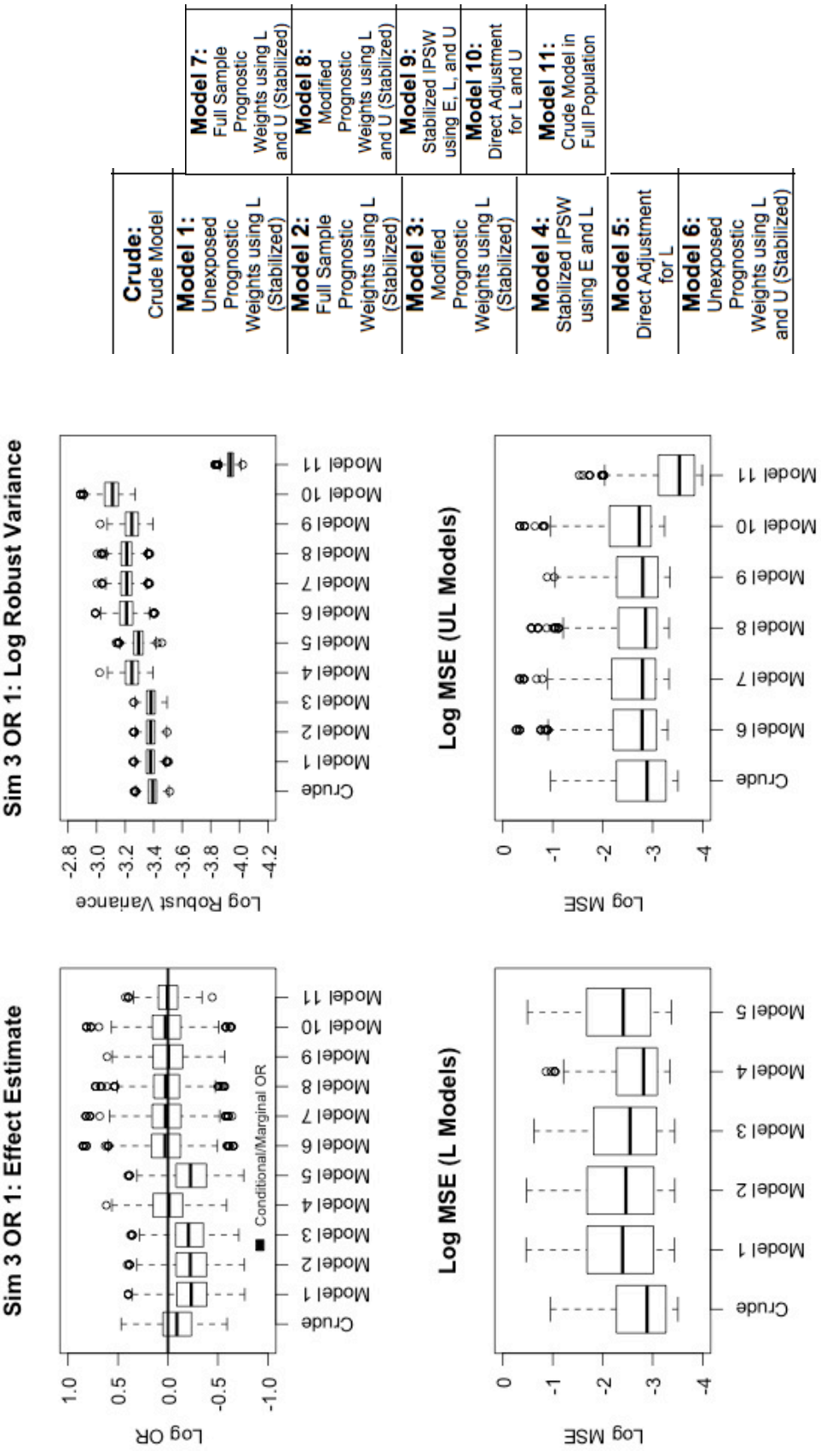


Figure 24: Simulation 3 (Conditional OR=1) Results

Table 17: Results for Simulation 3, OR=1

Description	Median logOR [95% CI Bootstrap]	Median Robust Variance	Median MSE (Conditional)	Monte Carlo Variance (bootstrap)
Crude: Crude Model	-0.08912 [-0.45, 0.34]	0.03357	0.05571	0.040208
Model 1: Unexposed Prognostic Weights using L (Stabilized)	-0.23270 [-0.61, 0.22]	0.03413	0.09108	0.045303
Model 2: Full Sample Prognostic Weights using L (Stabilized)	-0.22373 [-0.63, 0.21]	0.03407	0.08522	0.045418
Model 3: Modified Prognostic Weights using L (Stabilized)	-0.20548 [-0.57, 0.19]	0.03400	0.07838	0.038154
Model 4: Stabilized IPSW using E and L	-0.010761 [-0.42, 0.42]	0.03888	0.05983	0.045282
Model 5: Direct Adjustment for L	-0.22540 [-0.63, 0.21]	0.03710	0.08979	0.045483
Model 6: Unexposed Prognostic Weights using L and U (Stabilized)	0.02936 [-0.40, 0.50]	0.04037	0.06160	0.054761
Model 7: Full Sample Prognostic Weights using L and U (Stabilized)	0.02569 [-0.42, 0.50]	0.04031	0.06100	0.053531
Model 8: Modified Prognostic Weights using L and U (Stabilized)	0.02473 [-0.37, 0.46]	0.04032	0.05741	0.043175
Model 9: Stabilized IPSW using E, L, and U	-0.005906 [-0.41, 0.43]	0.03890	0.06076	0.045513
Model 10: Direct Adjustment for L and U	0.02649 [-0.42, 0.49]	0.04454	0.06516	0.053304
Model 11: Crude Model in Full Population	0.005759 [-0.27, 0.26]	0.01948	0.02918	0.018411

3.4 Discussion

This chapter examined the use of prognostic score weighting as a method to address selection bias. Three variants of the prognostic score were incorporated into several simulations: the prognostic score estimated in the unexposed group, the prognostic score estimated in the full population with weights based on predicted values where the observations are fixed to be unexposed, and a modified prognostic score where the outcome is modeled as a function of the covariates without regard to the exposure. It was hypothesized that the three prognostic models would perform similarly and return an approximately unbiased effect estimate. In all three simulations when the conditional OR was set to 3, the unexposed prognostic score weights and full population prognostic score weights generated an estimate of the conditional odds ratio while weighting using the inverse probability of selection generated marginal odds ratio. The modified prognostic score appeared to result in an estimate somewhere between the marginal and conditional OR for Simulations 2 and 3, while it was closer to the conditional OR for Simulation 1. In Hansen's 2008 publication on the prognostic score, he mentions that for a prognostic score that is estimated in both the unexposed and exposed group and the exposure increased the outcome there is a potential that "the estimated prognosis will be a mixture of the true propensity and prognostic scores."[2] This is referring to the full sample prognostic score; however, it could provide some insight into the observed results for the modified prognostic score. More information on how the

various prognostic score approaches perform when accounting for different biases is needed.

This chapter also assessed the performance of different methods combining prognostic scores and inverse probability weighting to correct for both selection bias and confounding. The combination weights performed comparably to other methods, though the three prognostic weights based on the selection and confounding variables had a lower median percent relative bias. A better understanding of the methods that can address the combination of selection bias and confounding are needed. Further studies should examine the use of the prognostic score as an approach to multiple biases, including selection bias.

Finally, Simulation 3 examined the case of selection bias where a variable causing selection is a collider with an unmeasured variable inducing the selection bias. Without including the unmeasured variable in the model or in the model for the weights, the only approach that was able to remove the selection bias was the inverse probability of selection weights (IPSW) based on the exposure and the measured selection collider. Direct adjustment on the measured selection collider and the three prognostic score weights based on the measured selection variables induced more bias into the model. While IPSW performed well, it requires that information be known about those who were not selected into the study in order to estimate the probability of selection. This is a practical approach for nested studies; however, outside of a larger study, the data is difficult to obtain. Further methods to address immigrative selection bias particularly when there is an unmeasured variable inducing selection or a collider are needed.

Chapter 4: Summary and Discussion

The prognostic score has been found to be an acceptable alternative to the use of the propensity score as a covariate balancing score and can appropriately remove bias due to confounding.[2-4] The prognostic score has also been shown to be a promising supplementary approach to confounding in combination with the propensity score.[5,9] This thesis sought to examine potential applications of the prognostic score to confounding that could prove to be equivalent to the propensity score with the aim that the applications would result in a dual approach to address confounding. One objective was to assess the equivalency and performance of the modified prognostic score and the propensity score in a simulation of a confounded logistic model with binary exposure. Methods combining the modified prognostic score and the propensity score were also assessed. The results indicated that weighting using either the modified prognostic score or the propensity score removed bias due to confounding. It was found that both scores yielded the marginal odds ratio, whereas the standard prognostic score restricted to the unexposed for developing the model or the full sample prognostic score resulted in a conditional OR. While weighting on the modified prognostic or propensity score yielded the marginal odds ratio, the combination methods returned estimates closer to the conditional odds ratio.

Methods for immigrative selection bias are often limited by the availability of data for those who did not enter the study. This thesis examined the use of the prognostic score as a potential approach to immigrative selection bias. Three

variants of the prognostic score were incorporated into several simulations: unexposed prognostic score, the full population prognostic score, and the modified prognostic score described in Chapter 2. In Simulations 1, 2, and 3, the unexposed and full population prognostic weights generated an estimate of the conditional odds ratio. In Simulations 2 and 3, the modified prognostic score yielded an estimate that appeared to be a mix of the marginal and conditional OR. The modified prognostic score yielded an estimate similar to the conditional OR in Simulation 1, but the percent relative bias was higher than the other prognostic models and it is likely that the effect estimate could have been a mixture of the conditional and marginal OR. As mentioned in Chapter 3, Hansen refers to this possibility for the full sample prognostic score, so perhaps the result for the modified prognostic score, which includes the full sample and does not account for the exposure can in part be explained. This result contrasted the marginal odds ratio that was obtained in the confounding simulation. Perhaps the performance of the score is influenced by the underlying causal structure in selection bias, which resulted in an estimate of the conditional odds ratio rather than the marginal odds ratio that was obtained in the confounding simulation. The combination of IPW and prognostic weights to address simultaneous selection bias and confounding performed comparably to other methods. The case of selection bias where a variable causing selection is a collider with an unmeasured variable inducing the selection bias was also examined. Inverse probability of selection weights (IPSW) based on the exposure and the measured selection collider yielded an unbiased effect estimate while the prognostic score

approaches induced more bias.

The use of a modified prognostic score and the application of prognostic scores to selection biased increased the understanding of the performance and potential applications of prognostic scores to epidemiologic models. The differing estimation of the marginal and conditional odds ratio while using the various prognostic score approaches should be examined further. Additionally, more information regarding the use of weighting in combined prognostic and propensity score methods is needed. The performance of the combined weights differed by causal model, including whether the effect estimated the marginal or conditional odds ratio. Further methods to address immigrative selection bias that do not rely on information on those outside of the study population. Similarly, methods for identifying and developing a causal model of selection bias are needed to increase understanding and methodological interest in selection bias.

Appendix A:

This is the R code for the Confounding Simulation detailed in Chapter 2. Part 1 is the code for when the conditional odds ratio is set to 3, and Part 2 is the code for when the conditional odds ratio is set to 1.

```
#############################################################################
#                     Part 1: Simulation based on Confounding DAG:          #
#                  Compares Prognostic and Propensity Weight                #
#                           Approaches to Confounding                       #
#############################################################################

conf.dat<-function(n, beta4, seed){
      set.seed(seed)

      Cvar1<-rbinom(n, 1, 0.5)
      Cvar2<-rbinom(n, 1, 0.6)
      Cvar3<-rbinom(n, 1, 0.5)

      delta0<-log(.3/.7)
      delta1<-log(1/3)
      delta2<-log(3)
      delta3<-log(3)
      probE<-exp(delta0+delta1*Cvar1+delta2*Cvar2+delta3*Cvar3)/(1 +
                  exp(delta0+delta1*Cvar1+delta2*Cvar2+delta3*Cvar3))
      Evar<-rbinom(n,1, prob= probE)
      beta0<-log(.3/.7)
      beta1<-log(3)
      beta2<-log(3)
      beta3<-log(3)
      probY<-exp(beta0+beta1*Cvar1+beta2*Cvar2+beta3*Cvar3+beta4*Evar)/(1 +
                  exp(beta0+beta1*Cvar1+beta2*Cvar2+beta3*Cvar3+beta4*Evar))
      Yvar<-rbinom(n, 1, prob=probY)
      dat.conf<-data.frame(Evar, Yvar, Cvar1, Cvar2, Cvar3)

      # crude
      crude.y.conf<-glm(Yvar~Evar,data=dat.conf,family=binomial(link="logit"))
      crude.e.conf<-glm(Evar~Yvar,data=dat.conf,family=binomial(link="logit"))

      # model 1: unexposed prognostic score
      wt1.conf <- glm(Yvar~Cvar1 + Cvar2 + Cvar3,
                      data=dat.conf[dat.conf$Evar==0,],family=binomial)
      dat.conf$wt1.conf <- predict(wt1.conf,newdata=dat.conf,type="response")
      dat.conf$wt1.conf[dat.conf$Yvar==0]<-
             1-dat.conf$wt1.conf[dat.conf$Yvar==0]
      wt1<-1/(dat.conf$wt1.conf)
      wt1.s <- (dat.conf$Yvar*mean(dat.conf$Yvar) +
                      (1-dat.conf$Yvar)*(1-mean(dat.conf$Yvar))) * wt1
      mod1y.conf<-glm(Yvar ~ Evar, data=dat.conf,
                      weights=wt1.s, family=binomial(link="logit"))
      mod1e.conf<-glm(Evar ~ Yvar, data=dat.conf,
                      weights=wt1.s, family=binomial(link="logit"))

      # model 2: full sample prognostic score

      wt2 <- glm(Yvar~Cvar1 + Cvar2 + Cvar3 +Evar,
```

```
       data=dat.conf,family=binomial)
tmp <- dat.conf
tmp$Evar[tmp$Evar==1] <- 0
wt2<- predict(wt2,newdata=tmp,type="response")
wt2.s <- dat.conf$Yvar*(mean(dat.conf$Yvar)/wt2) +
      (1-dat.conf$Yvar)*((1-mean(dat.conf$Yvar))/(1-wt2))
mod2y.conf<-glm(Yvar ~ Evar, data=dat.conf, weights=wt2.s,
                family=binomial(link="logit"))
mod2e.conf<-glm(Evar ~ Yvar, data=dat.conf, weights=wt2.s,
                family=binomial(link="logit"))


# model 3: modified prognostic score
wt3 <- glm(Yvar~Cvar1 + Cvar2 + Cvar3,
           data=dat.conf,family=binomial)$fitted.values
wt3.s <- dat.conf$Yvar*(mean(dat.conf$Yvar)/wt3) +
      (1-dat.conf$Yvar)*((1-mean(dat.conf$Yvar))/(1-wt3))
mod3y.conf<-glm(Yvar ~ Evar, data=dat.conf, weights=wt3.s,
                family=binomial(link="logit"))
mod3e.conf<-glm(Evar ~ Yvar, data=dat.conf, weights=wt3.s,
                family=binomial(link="logit"))

# model 4: stabilized IPEW
wt4 <- glm(Evar~Cvar1 + Cvar2 + Cvar3,data=dat.conf,
           family=binomial)$fitted.values
wt4.s <- dat.conf$Evar*(mean(dat.conf$Evar)/wt4) +
      (1-dat.conf$Evar)*((1-mean(dat.conf$Evar))/(1-wt4))
mod4y.conf<-glm(Yvar ~ Evar, data=dat.conf, weights=wt4.s,
                family=binomial(link="logit"))
mod4e.conf<-glm(Evar ~ Yvar, data=dat.conf, weights=wt4.s,
                family=binomial(link="logit"))

# model 5: direct adjustment
mod5y.conf<-glm(Yvar~Evar+Cvar1+Cvar2+Cvar3,data=dat.conf,
                family=binomial(link="logit"))
mod5e.conf<-glm(Evar~Yvar+Cvar1+Cvar2+Cvar3,data=dat.conf,
                family=binomial(link="logit"))

# model 6: combo weights-
 # modified prog score (stabilized) and IPEW (stabilized)
combo<-wt3.s*wt4.s
mod6y.conf<-glm(Yvar ~ Evar, data=dat.conf, weights=combo,
                family=binomial(link="logit"))
mod6e.conf<-glm(Evar ~ Yvar, data=dat.conf, weights=combo,
                family=binomial(link="logit"))

# weight by modified prog score, subclassification by propensity
library(MatchIt)
sub.set1 <- matchit(Evar ~ Cvar1 + Cvar2 + Cvar3,data=dat.conf,
            method="subclass",sub.by="all",subclass=5,discard="both")
sub.dat1 <- match.data(sub.set1,"all")
mod7y.conf <- glm(Yvar~Evar+factor(subclass),data=sub.dat1,
                  weights=wt3.s,family=binomial(link="logit"))
mod7e.conf <- glm(Evar~Yvar+factor(subclass),data=sub.dat1,
                  weights=wt3.s,family=binomial(link="logit"))

# weight by ipw (propensity), stratify by modified prog score
sub.set2 <- matchit(Yvar ~ Cvar1 + Cvar2 + Cvar3,data=dat.conf,
            method="subclass",sub.by="all",subclass=5,discard="both")
```

```
sub.dat2 <- match.data(sub.set2,"all")
mod8y.conf <- glm(Yvar~Evar+factor(subclass),data=sub.dat2,
                  weights=wt4.s,family=binomial(link="logit"))
mod8e.conf <- glm(Evar~Yvar+factor(subclass),data=sub.dat2,
                  weights=wt4.s,family=binomial(link="logit"))

# marginal or (for y on e)
pbar0 <- (beta0+beta1*Cvar1+beta2*Cvar2+beta3*Cvar3)
pbar0 <- mean(1/(1+exp(-pbar0)))
pbar1 <- (beta0+beta1*Cvar1+beta2*Cvar2+beta3*Cvar3+beta4)
pbar1 <- mean(1/(1+exp(-pbar1)))
marg.or <- (pbar1/(1-pbar1)) / (pbar0/(1-pbar0))

beta.crude.y<-summary(crude.y.conf)$coef[2,1]
var.crude.y<-(summary(crude.y.conf)$coef[2,2])^2
beta.crude.e<-summary(crude.e.conf)$coef[2,1]
var.crude.e<-(summary(crude.e.conf)$coef[2,2])^2
beta.mod1y<-summary(mod1y.conf)$coef[2,1]
var.mod1y<-(summary(mod1y.conf)$coef[2,2])^2
beta.mod1e<-summary(mod1e.conf)$coef[2,1]
var.mod1e<-(summary(mod1e.conf)$coef[2,2])^2
beta.mod2y<-summary(mod2y.conf)$coef[2,1]
var.mod2y<-(summary(mod2y.conf)$coef[2,2])^2
beta.mod2e<-summary(mod2e.conf)$coef[2,1]
var.mod2e<-(summary(mod2e.conf)$coef[2,2])^2
beta.mod3y<-summary(mod3y.conf)$coef[2,1]
var.mod3y<-(summary(mod3y.conf)$coef[2,2])^2
beta.mod3e<-summary(mod3e.conf)$coef[2,1]
var.mod3e<-(summary(mod3e.conf)$coef[2,2])^2
beta.mod4y<-summary(mod4y.conf)$coef[2,1]
var.mod4y<-(summary(mod4y.conf)$coef[2,2])^2
beta.mod4e<-summary(mod4e.conf)$coef[2,1]
var.mod4e<-(summary(mod4e.conf)$coef[2,2])^2
beta.mod5y<-summary(mod5y.conf)$coef[2,1]
var.mod5y<-(summary(mod5y.conf)$coef[2,2])^2
beta.mod5e<-summary(mod5e.conf)$coef[2,1]
var.mod5e<-(summary(mod5e.conf)$coef[2,2])^2
beta.mod6y<-summary(mod6y.conf)$coef[2,1]
var.mod6y<-(summary(mod6y.conf)$coef[2,2])^2
beta.mod6e<-summary(mod6e.conf)$coef[2,1]
var.mod6e<-(summary(mod6e.conf)$coef[2,2])^2
beta.mod7y<-summary(mod7y.conf)$coef[2,1]
var.mod7y<-(summary(mod7y.conf)$coef[2,2])^2
beta.mod7e<-summary(mod7e.conf)$coef[2,1]
var.mod7e<-(summary(mod7e.conf)$coef[2,2])^2
beta.mod8y<-summary(mod8y.conf)$coef[2,1]
var.mod8y<-(summary(mod8y.conf)$coef[2,2])^2
beta.mod8e<-summary(mod8e.conf)$coef[2,1]
var.mod8e<-(summary(mod8e.conf)$coef[2,2])^2

## insert calculation for % bias
bias.crudecond.y=((beta.crude.y-log(3))/log(3))*100
bias.crudemarg.y=((beta.crude.y-log(marg.or))/log(marg.or))*100
bias.crudecond.e=((beta.crude.e-log(3))/log(3))*100
bias.crudemarg.e=((beta.crude.e-log(marg.or))/log(marg.or))*100
bias.mod1y.c=((beta.mod1y-log(3))/log(3))*100
bias.mod1y.m=((beta.mod1y-log(marg.or))/log(marg.or))*100
bias.mod1e.c=((beta.mod1e-log(3))/log(3))*100
bias.mod1e.m=((beta.mod1e-log(marg.or))/log(marg.or))*100
```

```
bias.mod2y.c=((beta.mod2y-log(3))/log(3))*100
bias.mod2y.m=((beta.mod2y-log(marg.or))/log(marg.or))*100
bias.mod2e.c=((beta.mod2e-log(3))/log(3))*100
bias.mod2e.m=((beta.mod2e-log(marg.or))/log(marg.or))*100
bias.mod3y.c=((beta.mod3y-log(3))/log(3))*100
bias.mod3y.m=((beta.mod3y-log(marg.or))/log(marg.or))*100
bias.mod3e.c=((beta.mod3e-log(3))/log(3))*100
bias.mod3e.m=((beta.mod3e-log(marg.or))/log(marg.or))*100
bias.mod4y.c=((beta.mod4y-log(3))/log(3))*100
bias.mod4y.m=((beta.mod4y-log(marg.or))/log(marg.or))*100
bias.mod4e.c=((beta.mod4e-log(3))/log(3))*100
bias.mod4e.m=((beta.mod4e-log(marg.or))/log(marg.or))*100
bias.mod5y.c=((beta.mod5y-log(3))/log(3))*100
bias.mod5y.m=((beta.mod5y-log(marg.or))/log(marg.or))*100
bias.mod5e.c=((beta.mod5e-log(3))/log(3))*100
bias.mod5e.m=((beta.mod5e-log(marg.or))/log(marg.or))*100
bias.mod6y.c=((beta.mod6y-log(3))/log(3))*100
bias.mod6y.m=((beta.mod6y-log(marg.or))/log(marg.or))*100
bias.mod6e.c=((beta.mod6e-log(3))/log(3))*100
bias.mod6e.m=((beta.mod6e-log(marg.or))/log(marg.or))*100
bias.mod7y.c=((beta.mod7y-log(3))/log(3))*100
bias.mod7y.m=((beta.mod7y-log(marg.or))/log(marg.or))*100
bias.mod7e.c=((beta.mod7e-log(3))/log(3))*100
bias.mod7e.m=((beta.mod7e-log(marg.or))/log(marg.or))*100
bias.mod8y.c=((beta.mod8y-log(3))/log(3))*100
bias.mod8y.m=((beta.mod8y-log(marg.or))/log(marg.or))*100
bias.mod8e.c=((beta.mod8e-log(3))/log(3))*100
bias.mod8e.m=((beta.mod8e-log(marg.or))/log(marg.or))*100

## robust variance
library(sandwich)
robust.crude.y <- diag(sandwich(crude.y.conf))
robust.crude.y <- robust.crude.y[2]
robust.crude.e <- diag(sandwich(crude.e.conf))
robust.crude.e <- robust.crude.e[2]
robust.mod1y <- diag(sandwich(mod1y.conf))
robust.mod1y <-robust.mod1y[2]
robust.mod1e <- diag(sandwich(mod1e.conf))
robust.mod1e <-robust.mod1e[2]
robust.mod2y <- diag(sandwich(mod2y.conf))
robust.mod2y <-robust.mod2y[2]
robust.mod2e <- diag(sandwich(mod2e.conf))
robust.mod2e <-robust.mod2e[2]
robust.mod3y <- diag(sandwich(mod3y.conf))
robust.mod3y <-robust.mod3y[2]
robust.mod3e <- diag(sandwich(mod3e.conf))
robust.mod3e <-robust.mod3e[2]
robust.mod4y <- diag(sandwich(mod4y.conf))
robust.mod4y <-robust.mod4y[2]
robust.mod4e <- diag(sandwich(mod4e.conf))
robust.mod4e <-robust.mod4e[2]
robust.mod5y <- diag(sandwich(mod5y.conf))
robust.mod5y <-robust.mod5y[2]
robust.mod5e <- diag(sandwich(mod5e.conf))
robust.mod5e <-robust.mod5e[2]
robust.mod6y <- diag(sandwich(mod6y.conf))
robust.mod6y <-robust.mod6y[2]
robust.mod6e <- diag(sandwich(mod6e.conf))
robust.mod6e <-robust.mod6e[2]
```

```
robust.mod7y <- diag(sandwich(mod7y.conf))
robust.mod7y <-robust.mod7y[2]
robust.mod7e <- diag(sandwich(mod7e.conf))
robust.mod7e <-robust.mod7e[2]
robust.mod8y <- diag(sandwich(mod8y.conf))
robust.mod8y <-robust.mod8y[2]
robust.mod8e <- diag(sandwich(mod8e.conf))
robust.mod8e <-robust.mod8e[2]

## mse calculation on log scale (using robust variance)
mse.crudecond.y=((bias.crudecond.y/100)^2)+robust.crude.y
mse.crudemarg.y=((bias.crudemarg.y/100)^2)+robust.crude.y
mse.crudecond.e=((bias.crudecond.e/100)^2)+robust.crude.e
mse.crudemarg.e=((bias.crudemarg.e/100)^2)+robust.crude.e
mse.mod1y.c=((bias.mod1y.c/100)^2)+robust.mod1y
mse.mod1e.c=((bias.mod1e.c/100)^2)+robust.mod1e
mse.mod1y.m=((bias.mod1y.m/100)^2)+robust.mod1y
mse.mod1e.m=((bias.mod1e.m/100)^2)+robust.mod1e
mse.mod2y.c=((bias.mod2y.c/100)^2)+robust.mod2y
mse.mod2e.c=((bias.mod2e.c/100)^2)+robust.mod2e
mse.mod2y.m=((bias.mod2y.m/100)^2)+robust.mod2y
mse.mod2e.m=((bias.mod2e.m/100)^2)+robust.mod2e
mse.mod3y.c=((bias.mod3y.c/100)^2)+robust.mod3y
mse.mod3e.c=((bias.mod3e.c/100)^2)+robust.mod3e
mse.mod3y.m=((bias.mod3y.m/100)^2)+robust.mod3y
mse.mod3e.m=((bias.mod3e.m/100)^2)+robust.mod3e
mse.mod4y.c=((bias.mod4y.c/100)^2)+robust.mod4y
mse.mod4e.c=((bias.mod4e.c/100)^2)+robust.mod4e
mse.mod4y.m=((bias.mod4y.m/100)^2)+robust.mod4y
mse.mod4e.m=((bias.mod4e.m/100)^2)+robust.mod4e
mse.mod5y.c=((bias.mod5y.c/100)^2)+robust.mod5y
mse.mod5e.c=((bias.mod5e.c/100)^2)+robust.mod5e
mse.mod5y.m=((bias.mod5y.m/100)^2)+robust.mod5y
mse.mod5e.m=((bias.mod5e.m/100)^2)+robust.mod5e
mse.mod6y.c=((bias.mod6y.c/100)^2)+robust.mod6y
mse.mod6e.c=((bias.mod6e.c/100)^2)+robust.mod6e
mse.mod6y.m=((bias.mod6y.m/100)^2)+robust.mod6y
mse.mod6e.m=((bias.mod6e.m/100)^2)+robust.mod6e
mse.mod7y.c=((bias.mod7y.c/100)^2)+robust.mod7y
mse.mod7e.c=((bias.mod7e.c/100)^2)+robust.mod7e
mse.mod7y.m=((bias.mod7y.m/100)^2)+robust.mod7y
mse.mod7e.m=((bias.mod7e.m/100)^2)+robust.mod7e
mse.mod8y.c=((bias.mod8y.c/100)^2)+robust.mod8y
mse.mod8e.c=((bias.mod8e.c/100)^2)+robust.mod8e
mse.mod8y.m=((bias.mod8y.m/100)^2)+robust.mod8y
mse.mod8e.m=((bias.mod8e.m/100)^2)+robust.mod8e

lmse.crudecond.y=log(((bias.crudecond.y/100)^2)+robust.crude.y)
lmse.crudemarg.y=log(((bias.crudemarg.y/100)^2)+robust.crude.y)
lmse.crudecond.e=log(((bias.crudecond.e/100)^2)+robust.crude.e)
lmse.crudemarg.e=log(((bias.crudemarg.e/100)^2)+robust.crude.e)
lmse.mod1y.c=log(((bias.mod1y.c/100)^2)+robust.mod1y)
lmse.mod1e.c=log(((bias.mod1e.c/100)^2)+robust.mod1e)
lmse.mod1y.m=log(((bias.mod1y.m/100)^2)+robust.mod1y)
lmse.mod1e.m=log(((bias.mod1e.m/100)^2)+robust.mod1e)
lmse.mod2y.c=log(((bias.mod2y.c/100)^2)+robust.mod2y)
lmse.mod2e.c=log(((bias.mod2e.c/100)^2)+robust.mod2e)
lmse.mod2y.m=log(((bias.mod2y.m/100)^2)+robust.mod2y)
lmse.mod2e.m=log(((bias.mod2e.m/100)^2)+robust.mod2e)
```

```
lmse.mod3y.c=log(((bias.mod3y.c/100)^2)+robust.mod3y)
lmse.mod3e.c=log(((bias.mod3e.c/100)^2)+robust.mod3e)
lmse.mod3y.m=log(((bias.mod3y.m/100)^2)+robust.mod3y)
lmse.mod3e.m=log(((bias.mod3e.m/100)^2)+robust.mod3e)
lmse.mod4y.c=log(((bias.mod4y.c/100)^2)+robust.mod4y)
lmse.mod4e.c=log(((bias.mod4e.c/100)^2)+robust.mod4e)
lmse.mod4y.m=log(((bias.mod4y.m/100)^2)+robust.mod4y)
lmse.mod4e.m=log(((bias.mod4e.m/100)^2)+robust.mod4e)
lmse.mod5y.c=log(((bias.mod5y.c/100)^2)+robust.mod5y)
lmse.mod5e.c=log(((bias.mod5e.c/100)^2)+robust.mod5e)
lmse.mod5y.m=log(((bias.mod5y.m/100)^2)+robust.mod5y)
lmse.mod5e.m=log(((bias.mod5e.m/100)^2)+robust.mod5e)
lmse.mod6y.c=log(((bias.mod6y.c/100)^2)+robust.mod6y)
lmse.mod6e.c=log(((bias.mod6e.c/100)^2)+robust.mod6e)
lmse.mod6y.m=log(((bias.mod6y.m/100)^2)+robust.mod6y)
lmse.mod6e.m=log(((bias.mod6e.m/100)^2)+robust.mod6e)
lmse.mod7y.c=log(((bias.mod7y.c/100)^2)+robust.mod7y)
lmse.mod7e.c=log(((bias.mod7e.c/100)^2)+robust.mod7e)
lmse.mod7y.m=log(((bias.mod7y.m/100)^2)+robust.mod7y)
lmse.mod7e.m=log(((bias.mod7e.m/100)^2)+robust.mod7e)
lmse.mod8y.c=log(((bias.mod8y.c/100)^2)+robust.mod8y)
lmse.mod8e.c=log(((bias.mod8e.c/100)^2)+robust.mod8e)
lmse.mod8y.m=log(((bias.mod8y.m/100)^2)+robust.mod8y)
lmse.mod8e.m=log(((bias.mod8e.m/100)^2)+robust.mod8e)

### convert robust variance to log scale

lrobust.crude.y <- log(robust.crude.y)
lrobust.crude.e <- log(robust.crude.e)
lrobust.mod1y <-log(robust.mod1y)
lrobust.mod1e <-log(robust.mod1e)
lrobust.mod2y <-log(robust.mod2y)
lrobust.mod2e <-log(robust.mod2e)
lrobust.mod3y <-log(robust.mod3y)
lrobust.mod3e <-log(robust.mod3e)
lrobust.mod4y <-log(robust.mod4y)
lrobust.mod4e <-log(robust.mod4e)
lrobust.mod5y <-log(robust.mod5y)
lrobust.mod5e <-log(robust.mod5e)
lrobust.mod6y <-log(robust.mod6y)
lrobust.mod6e <-log(robust.mod6e)
lrobust.mod7y <-log(robust.mod7y)
lrobust.mod7e <-log(robust.mod7e)
lrobust.mod8y <-log(robust.mod8y)
lrobust.mod8e <-log(robust.mod8e)

res.frame<-data.frame(beta.crude.y, var.crude.y, beta.crude.e,
                      var.crude.e, beta.mod1y, var.mod1y, beta.mod1e,
                      var.mod1e, beta.mod2y, var.mod2y, beta.mod2e,
                      var.mod2e, beta.mod3y, var.mod3y, beta.mod3e,
                      var.mod3e, beta.mod4y, var.mod4y, beta.mod4e,
                      var.mod4e, beta.mod5y, var.mod5y, beta.mod5e,
                      var.mod5e, beta.mod6y, var.mod6y, beta.mod6e,
                      var.mod6e, beta.mod7y, var.mod7y, beta.mod7e,
                      var.mod7e, beta.mod8y, var.mod8y, beta.mod8e,
                      var.mod8e, bias.crudecond.y, bias.crudemarg.y,
                      bias.crudecond.e, bias.crudemarg.e, bias.mod1y.c,
                      bias.mod1y.m, bias.mod1e.c, bias.mod1e.m,
                      bias.mod2y.c, bias.mod2y.m, bias.mod2e.c,
```

```
                              bias.mod2e.m, bias.mod3y.c, bias.mod3y.m,
                              bias.mod3e.c, bias.mod3e.m, bias.mod4y.c,
                              bias.mod4y.m, bias.mod4e.c, bias.mod4e.m,
                              bias.mod5y.c, bias.mod5y.m, bias.mod5e.c,
                              bias.mod5e.m, bias.mod6y.c, bias.mod6y.m,
                              bias.mod6e.c, bias.mod6e.m, bias.mod7y.c,
                              bias.mod7y.m, bias.mod7e.c, bias.mod7e.m,
                              bias.mod8y.c, bias.mod8y.m, bias.mod8e.c,
                              bias.mod8e.m, robust.crude.y, robust.crude.e,
                              robust.mod1y, robust.mod1e, robust.mod2y,
                              robust.mod2e, robust.mod3y, robust.mod3e,
                              robust.mod4y, robust.mod4e, robust.mod5y,
                              robust.mod5e, robust.mod6y, robust.mod6e,
                              robust.mod7y, robust.mod7e, robust.mod8y,
                              robust.mod8e, mse.crudecond.y, mse.crudecond.e,
                              mse.crudemarg.y, mse.crudemarg.e, mse.mod1y.c,
                              mse.mod1y.m, mse.mod1e.c, mse.mod1e.m, mse.mod2y.c,
                              mse.mod2y.m, mse.mod2e.c, mse.mod2e.m, mse.mod3y.c,
                              mse.mod3y.m, mse.mod3e.c, mse.mod3e.m, mse.mod4y.c,
                              mse.mod4y.m, mse.mod4e.c, mse.mod4e.m, mse.mod5y.c,
                              mse.mod5y.m, mse.mod5e.c, mse.mod5e.m, mse.mod6y.c,
                              mse.mod6y.m, mse.mod6e.c, mse.mod6e.m, mse.mod7y.c,
                              mse.mod7y.m, mse.mod7e.c, mse.mod7e.m, mse.mod8y.c,
                              mse.mod8y.m, mse.mod8e.c, mse.mod8e.m,
                              lrobust.crude.y, lrobust.crude.e, lrobust.mod1y,
                              lrobust.mod1e, lrobust.mod2y, lrobust.mod2e,
                              lrobust.mod3y, lrobust.mod3e, lrobust.mod4y,
                              lrobust.mod4e, lrobust.mod5y, lrobust.mod5e,
                              lrobust.mod6y, lrobust.mod6e, lrobust.mod7y,
                              lrobust.mod7e, lrobust.mod8y, lrobust.mod8e,
                              lmse.crudecond.y,lmse.crudecond.e,lmse.crudemarg.y,
                              lmse.crudemarg.e, lmse.mod1y.c, lmse.mod1y.m,
                              lmse.mod1e.c, lmse.mod1e.m, lmse.mod2y.c,
                              lmse.mod2y.m, lmse.mod2e.c, lmse.mod2e.m,
                              lmse.mod3y.c, lmse.mod3y.m, lmse.mod3e.c,
                              lmse.mod3e.m, lmse.mod4y.c, lmse.mod4y.m,
                              lmse.mod4e.c, lmse.mod4e.m, lmse.mod5y.c,
                              lmse.mod5y.m, lmse.mod5e.c, lmse.mod5e.m,
                              lmse.mod6y.c, lmse.mod6y.m, lmse.mod6e.c,
                              lmse.mod6e.m, lmse.mod7y.c, lmse.mod7y.m,
                              lmse.mod7e.c, lmse.mod7e.m, lmse.mod8y.c,
                              lmse.mod8y.m, lmse.mod8e.c, lmse.mod8e.m, marg.or)
      res.frame
}

set.seed(7501345)

temp.conf <- matrix(NA, 1000, 1000)

output <- apply(temp.conf, 1, function(x)
conf.dat(1000,log(3),round(runif(1)*100000)))
conf.data<-do.call("rbind",output)


### bootstrap
### bootstrap CIs and log variance for each model (y on e)
quantile(conf.data[,1],c(0.025,0.975)) # crude y
crude.boot.y<-var(conf.data[,1])
crude.boot.y
```

```
quantile(conf.data[,3],c(0.025,0.975)) # crude e
crude.boot.e<-var(conf.data[,3])
crude.boot.e

quantile(conf.data[,5],c(0.025,0.975)) # mod 1 y
mod1.boot.y<-var(conf.data[,5])
mod1.boot.y
quantile(conf.data[,7],c(0.025,0.975)) # mod 1 e
mod1.boot.e<-var(conf.data[,7])
mod1.boot.e

quantile(conf.data[,9],c(0.025,0.975)) # mod 2 y
mod2.boot.y<-var(conf.data[,9])
mod2.boot.y
quantile(conf.data[,11],c(0.025,0.975)) # mod 2 e
mod2.boot.e<-var(conf.data[,11])
mod2.boot.e

quantile(conf.data[,13],c(0.025,0.975)) # mod 3 y
mod3.boot.y<-var(conf.data[,13])
mod3.boot.y
quantile(conf.data[,15],c(0.025,0.975)) # mod 3 e
mod3.boot.e<-var(conf.data[,15])
mod3.boot.e

quantile(conf.data[,17],c(0.025,0.975)) # mod 4 y
mod4.boot.y<-var(conf.data[,17])
mod4.boot.y
quantile(conf.data[,19],c(0.025,0.975)) # mod 4 e
mod4.boot.e<-var(conf.data[,19])
mod4.boot.e

quantile(conf.data[,21],c(0.025,0.975)) # mod 5 y
mod5.boot.y<-var(conf.data[,21])
mod5.boot.y
quantile(conf.data[,23],c(0.025,0.975)) # mod 5 e
mod5.boot.e<-var(conf.data[,23])
mod5.boot.e

quantile(conf.data[,25],c(0.025,0.975)) # mod 6 y
mod6.boot.y<-var(conf.data[,25])
mod6.boot.y
quantile(conf.data[,27],c(0.025,0.975)) # mod 6 e
mod6.boot.e<-var(conf.data[,27])
mod6.boot.e

quantile(conf.data[,29],c(0.025,0.975)) # mod 7 y
mod7.boot.y<-var(conf.data[,29])
mod7.boot.y
quantile(conf.data[,31],c(0.025,0.975)) # mod 7 e
mod7.boot.e<-var(conf.data[,31])
mod7.boot.e

quantile(conf.data[,33],c(0.025,0.975)) # mod 8 y
mod8.boot.y<-var(conf.data[,33])
mod8.boot.y
quantile(conf.data[,35],c(0.025,0.975)) # mod 8 e
mod8.boot.e<-var(conf.data[,35])
mod8.boot.e
```

```
#############################################################################
#                     Part 2: Simulation based on Confounding DAG:          #
#                  Compares Prognostic and Propensity Weight                #
#                           Approaches to Confounding                       #
#############################################################################

conf.dat<-function(n, beta4, seed){
      set.seed(seed)

      Cvar1<-rbinom(n, 1, 0.5)
      Cvar2<-rbinom(n, 1, 0.6)
      Cvar3<-rbinom(n, 1, 0.5)

      delta0<-log(.3/.7)
      delta1<-log(1/3)
      delta2<-log(3)
      delta3<-log(3)
      probE<-exp(delta0+delta1*Cvar1+delta2*Cvar2+delta3*Cvar3)/
            (1 + exp(delta0+delta1*Cvar1+delta2*Cvar2+delta3*Cvar3))
      Evar<-rbinom(n,1, prob= probE)
      beta0<-log(.3/.7)
      beta1<-log(3)
      beta2<-log(3)
      beta3<-log(3)
      probY<-exp(beta0+beta1*Cvar1+beta2*Cvar2+beta3*Cvar3+beta4*Evar)/
            (1 + exp(beta0+beta1*Cvar1+beta2*Cvar2+beta3*Cvar3+beta4*Evar))
      Yvar<-rbinom(n, 1, prob=probY)
      dat.conf<-data.frame(Evar, Yvar, Cvar1, Cvar2, Cvar3)

      # crude
      crude.y.conf<-glm(Yvar~Evar,data=dat.conf,family=binomial(link="logit"))
      crude.e.conf<-glm(Evar~Yvar,data=dat.conf,family=binomial(link="logit"))

      # model 1: unexposed prognostic score
      wt1.conf <- glm(Yvar~Cvar1 + Cvar2 + Cvar3,
                      data=dat.conf[dat.conf$Evar==0,],family=binomial)
      dat.conf$wt1.conf <- predict(wt1.conf,newdata=dat.conf,type="response")
      dat.conf$wt1.conf[dat.conf$Yvar==0]<-
             1-dat.conf$wt1.conf[dat.conf$Yvar==0]
      wt1<-1/(dat.conf$wt1.conf)
      wt1.s <- (dat.conf$Yvar*mean(dat.conf$Yvar) +
                      (1-dat.conf$Yvar)*(1-mean(dat.conf$Yvar))) * wt1
      mod1y.conf<-glm(Yvar ~ Evar, data=dat.conf, weights=wt1.s,
                      family=binomial(link="logit"))
      mod1e.conf<-glm(Evar ~ Yvar, data=dat.conf, weights=wt1.s,
                      family=binomial(link="logit"))

      # model 2: full sample prognostic score

      wt2 <- glm(Yvar~Cvar1 + Cvar2 + Cvar3 +Evar,
             data=dat.conf,family=binomial)
      tmp <- dat.conf
      tmp$Evar[tmp$Evar==1] <- 0
      wt2<- predict(wt2,newdata=tmp,type="response")
      wt2.s <- dat.conf$Yvar*(mean(dat.conf$Yvar)/wt2) +
```

```
      (1-dat.conf$Yvar)*((1-mean(dat.conf$Yvar))/(1-wt2))
mod2y.conf<-glm(Yvar ~ Evar, data=dat.conf, weights=wt2.s,
                family=binomial(link="logit"))
mod2e.conf<-glm(Evar ~ Yvar, data=dat.conf, weights=wt2.s,
                family=binomial(link="logit"))


# model 3: modified prognostic score
wt3 <- glm(Yvar~Cvar1 + Cvar2 + Cvar3,data=dat.conf,
           family=binomial)$fitted.values
wt3.s <- dat.conf$Yvar*(mean(dat.conf$Yvar)/wt3) +
      (1-dat.conf$Yvar)*((1-mean(dat.conf$Yvar))/(1-wt3))
mod3y.conf<-glm(Yvar ~ Evar, data=dat.conf, weights=wt3.s,
                family=binomial(link="logit"))
mod3e.conf<-glm(Evar ~ Yvar, data=dat.conf, weights=wt3.s,
                family=binomial(link="logit"))

# model 4: stabilized IPEW
wt4 <- glm(Evar~Cvar1 + Cvar2 + Cvar3,data=dat.conf,
           family=binomial)$fitted.values
wt4.s <- dat.conf$Evar*(mean(dat.conf$Evar)/wt4) +
      (1-dat.conf$Evar)*((1-mean(dat.conf$Evar))/(1-wt4))
mod4y.conf<-glm(Yvar ~ Evar, data=dat.conf, weights=wt4.s,
                family=binomial(link="logit"))
mod4e.conf<-glm(Evar ~ Yvar, data=dat.conf, weights=wt4.s,
                family=binomial(link="logit"))

# model 5: direct adjustment
mod5y.conf<-glm(Yvar~Evar+Cvar1+Cvar2+Cvar3,data=dat.conf,
                family=binomial(link="logit"))
mod5e.conf<-glm(Evar~Yvar+Cvar1+Cvar2+Cvar3,data=dat.conf,
                family=binomial(link="logit"))

# model 6: combo weights-
 # modified prog score (stabilized) and IPEW (stabilized)
combo<-wt3.s*wt4.s
mod6y.conf<-glm(Yvar ~ Evar, data=dat.conf, weights=combo,
                family=binomial(link="logit"))
mod6e.conf<-glm(Evar ~ Yvar, data=dat.conf, weights=combo,
                family=binomial(link="logit"))

# weight by modified prog score, subclassification by propensity
library(MatchIt)
sub.set1 <- matchit(Evar ~ Cvar1 + Cvar2 + Cvar3,data=dat.conf,
            method="subclass",sub.by="all",subclass=5,discard="both")
sub.dat1 <- match.data(sub.set1,"all")
mod7y.conf <- glm(Yvar~Evar+factor(subclass),data=sub.dat1,weights=wt3.s,
                  family=binomial(link="logit"))
mod7e.conf <- glm(Evar~Yvar+factor(subclass),data=sub.dat1,weights=wt3.s,
                  family=binomial(link="logit"))

# weight by ipw (propensity), stratify by modified prog score
sub.set2 <- matchit(Yvar ~ Cvar1 + Cvar2 + Cvar3,data=dat.conf,
            method="subclass",sub.by="all",subclass=5,discard="both")
sub.dat2 <- match.data(sub.set2,"all")
mod8y.conf <- glm(Yvar~Evar+factor(subclass),data=sub.dat2,weights=wt4.s,
                  family=binomial(link="logit"))
mod8e.conf <- glm(Evar~Yvar+factor(subclass),data=sub.dat2,weights=wt4.s,
                  family=binomial(link="logit"))
```

```
# marginal or (for y on e)
pbar0 <- (beta0+beta1*Cvar1+beta2*Cvar2+beta3*Cvar3)
pbar0 <- mean(1/(1+exp(-pbar0)))
pbar1 <- (beta0+beta1*Cvar1+beta2*Cvar2+beta3*Cvar3+beta4)
pbar1 <- mean(1/(1+exp(-pbar1)))
marg.or <- (pbar1/(1-pbar1)) / (pbar0/(1-pbar0))

beta.crude.y<-summary(crude.y.conf)$coef[2,1]
var.crude.y<-(summary(crude.y.conf)$coef[2,2])^2
beta.crude.e<-summary(crude.e.conf)$coef[2,1]
var.crude.e<-(summary(crude.e.conf)$coef[2,2])^2
beta.mod1y<-summary(mod1y.conf)$coef[2,1]
var.mod1y<-(summary(mod1y.conf)$coef[2,2])^2
beta.mod1e<-summary(mod1e.conf)$coef[2,1]
var.mod1e<-(summary(mod1e.conf)$coef[2,2])^2
beta.mod2y<-summary(mod2y.conf)$coef[2,1]
var.mod2y<-(summary(mod2y.conf)$coef[2,2])^2
beta.mod2e<-summary(mod2e.conf)$coef[2,1]
var.mod2e<-(summary(mod2e.conf)$coef[2,2])^2
beta.mod3y<-summary(mod3y.conf)$coef[2,1]
var.mod3y<-(summary(mod3y.conf)$coef[2,2])^2
beta.mod3e<-summary(mod3e.conf)$coef[2,1]
var.mod3e<-(summary(mod3e.conf)$coef[2,2])^2
beta.mod4y<-summary(mod4y.conf)$coef[2,1]
var.mod4y<-(summary(mod4y.conf)$coef[2,2])^2
beta.mod4e<-summary(mod4e.conf)$coef[2,1]
var.mod4e<-(summary(mod4e.conf)$coef[2,2])^2
beta.mod5y<-summary(mod5y.conf)$coef[2,1]
var.mod5y<-(summary(mod5y.conf)$coef[2,2])^2
beta.mod5e<-summary(mod5e.conf)$coef[2,1]
var.mod5e<-(summary(mod5e.conf)$coef[2,2])^2
beta.mod6y<-summary(mod6y.conf)$coef[2,1]
var.mod6y<-(summary(mod6y.conf)$coef[2,2])^2
beta.mod6e<-summary(mod6e.conf)$coef[2,1]
var.mod6e<-(summary(mod6e.conf)$coef[2,2])^2
beta.mod7y<-summary(mod7y.conf)$coef[2,1]
var.mod7y<-(summary(mod7y.conf)$coef[2,2])^2
beta.mod7e<-summary(mod7e.conf)$coef[2,1]
var.mod7e<-(summary(mod7e.conf)$coef[2,2])^2
beta.mod8y<-summary(mod8y.conf)$coef[2,1]
var.mod8y<-(summary(mod8y.conf)$coef[2,2])^2
beta.mod8e<-summary(mod8e.conf)$coef[2,1]
var.mod8e<-(summary(mod8e.conf)$coef[2,2])^2

## robust variance
library(sandwich)
robust.crude.y <- diag(sandwich(crude.y.conf))
robust.crude.y <- robust.crude.y[2]
robust.crude.e <- diag(sandwich(crude.e.conf))
robust.crude.e <- robust.crude.e[2]
robust.mod1y <- diag(sandwich(mod1y.conf))
robust.mod1y <-robust.mod1y[2]
robust.mod1e <- diag(sandwich(mod1e.conf))
robust.mod1e <-robust.mod1e[2]
robust.mod2y <- diag(sandwich(mod2y.conf))
robust.mod2y <-robust.mod2y[2]
robust.mod2e <- diag(sandwich(mod2e.conf))
robust.mod2e <-robust.mod2e[2]
```

```
robust.mod3y <- diag(sandwich(mod3y.conf))
robust.mod3y <-robust.mod3y[2]
robust.mod3e <- diag(sandwich(mod3e.conf))
robust.mod3e <-robust.mod3e[2]
robust.mod4y <- diag(sandwich(mod4y.conf))
robust.mod4y <-robust.mod4y[2]
robust.mod4e <- diag(sandwich(mod4e.conf))
robust.mod4e <-robust.mod4e[2]
robust.mod5y <- diag(sandwich(mod5y.conf))
robust.mod5y <-robust.mod5y[2]
robust.mod5e <- diag(sandwich(mod5e.conf))
robust.mod5e <-robust.mod5e[2]
robust.mod6y <- diag(sandwich(mod6y.conf))
robust.mod6y <-robust.mod6y[2]
robust.mod6e <- diag(sandwich(mod6e.conf))
robust.mod6e <-robust.mod6e[2]
robust.mod7y <- diag(sandwich(mod7y.conf))
robust.mod7y <-robust.mod7y[2]
robust.mod7e <- diag(sandwich(mod7e.conf))
robust.mod7e <-robust.mod7e[2]
robust.mod8y <- diag(sandwich(mod8y.conf))
robust.mod8y <-robust.mod8y[2]
robust.mod8e <- diag(sandwich(mod8e.conf))
robust.mod8e <-robust.mod8e[2]

## mse calculation (using robust variance)
mse.crudecond.y=((beta.crude.y)^2)+robust.crude.y
mse.crudecond.e=((beta.crude.e)^2)+robust.crude.e
mse.mod1y.c=((beta.mod1y)^2)+robust.mod1y
mse.mod1e.c=((beta.mod1e)^2)+robust.mod1e
mse.mod2y.c=((beta.mod2y)^2)+robust.mod2y
mse.mod2e.c=((beta.mod2e)^2)+robust.mod2e
mse.mod3y.c=((beta.mod3y)^2)+robust.mod3y
mse.mod3e.c=((beta.mod3e)^2)+robust.mod3e
mse.mod4y.c=((beta.mod4y)^2)+robust.mod4y
mse.mod4e.c=((beta.mod4e)^2)+robust.mod4e
mse.mod5y.c=((beta.mod5y)^2)+robust.mod5y
mse.mod5e.c=((beta.mod5e)^2)+robust.mod5e
mse.mod6y.c=((beta.mod6y)^2)+robust.mod6y
mse.mod6e.c=((beta.mod6e)^2)+robust.mod6e
mse.mod7y.c=((beta.mod7y)^2)+robust.mod7y
mse.mod7e.c=((beta.mod7e)^2)+robust.mod7e
mse.mod8y.c=((beta.mod8y)^2)+robust.mod8y
mse.mod8e.c=((beta.mod8e)^2)+robust.mod8e

## mse calculation (using robust variance) on log scale
lmse.crudecond.y=log(((beta.crude.y)^2)+robust.crude.y)
lmse.crudecond.e=log(((beta.crude.e)^2)+robust.crude.e)
lmse.mod1y.c=log(((beta.mod1y)^2)+robust.mod1y)
lmse.mod1e.c=log(((beta.mod1e)^2)+robust.mod1e)
lmse.mod2y.c=log(((beta.mod2y)^2)+robust.mod2y)
lmse.mod2e.c=log(((beta.mod2e)^2)+robust.mod2e)
lmse.mod3y.c=log(((beta.mod3y)^2)+robust.mod3y)
lmse.mod3e.c=log(((beta.mod3e)^2)+robust.mod3e)
lmse.mod4y.c=log(((beta.mod4y)^2)+robust.mod4y)
lmse.mod4e.c=log(((beta.mod4e)^2)+robust.mod4e)
lmse.mod5y.c=log(((beta.mod5y)^2)+robust.mod5y)
lmse.mod5e.c=log(((beta.mod5e)^2)+robust.mod5e)
lmse.mod6y.c=log(((beta.mod6y)^2)+robust.mod6y)
```

```
lmse.mod6e.c=log(((beta.mod6e)^2)+robust.mod6e)
lmse.mod7y.c=log(((beta.mod7y)^2)+robust.mod7y)
lmse.mod7e.c=log(((beta.mod7e)^2)+robust.mod7e)
lmse.mod8y.c=log(((beta.mod8y)^2)+robust.mod8y)
lmse.mod8e.c=log(((beta.mod8e)^2)+robust.mod8e)

### convert robust variance to log scale

lrobust.crude.y <- log(robust.crude.y)
lrobust.crude.e <- log(robust.crude.e)
lrobust.mod1y <-log(robust.mod1y)
lrobust.mod1e <-log(robust.mod1e)
lrobust.mod2y <-log(robust.mod2y)
lrobust.mod2e <-log(robust.mod2e)
lrobust.mod3y <-log(robust.mod3y)
lrobust.mod3e <-log(robust.mod3e)
lrobust.mod4y <-log(robust.mod4y)
lrobust.mod4e <-log(robust.mod4e)
lrobust.mod5y <-log(robust.mod5y)
lrobust.mod5e <-log(robust.mod5e)
lrobust.mod6y <-log(robust.mod6y)
lrobust.mod6e <-log(robust.mod6e)
lrobust.mod7y <-log(robust.mod7y)
lrobust.mod7e <-log(robust.mod7e)
lrobust.mod8y <-log(robust.mod8y)
lrobust.mod8e <-log(robust.mod8e)

res.frame<-data.frame(beta.crude.y, var.crude.y, beta.crude.e,
                      var.crude.e, beta.mod1y, var.mod1y, beta.mod1e,
                      var.mod1e, beta.mod2y, var.mod2y, beta.mod2e,
                      var.mod2e, beta.mod3y, var.mod3y, beta.mod3e,
                      var.mod3e, beta.mod4y, var.mod4y, beta.mod4e,
                      var.mod4e, beta.mod5y, var.mod5y, beta.mod5e,
                      var.mod5e, beta.mod6y, var.mod6y, beta.mod6e,
                      var.mod6e, beta.mod7y, var.mod7y, beta.mod7e,
                      var.mod7e, beta.mod8y, var.mod8y, beta.mod8e,
                      var.mod8e, robust.crude.y, robust.crude.e,
                      robust.mod1y, robust.mod1e, robust.mod2y,
                      robust.mod2e, robust.mod3y, robust.mod3e,
                      robust.mod4y, robust.mod4e, robust.mod5y,
                      robust.mod5e, robust.mod6y, robust.mod6e,
                      robust.mod7y, robust.mod7e, robust.mod8y,
                      robust.mod8e, mse.crudecond.y, mse.crudecond.e,
                      mse.mod1y.c, mse.mod1e.c, mse.mod2y.c, mse.mod2e.c,
                      mse.mod3y.c, mse.mod3e.c, mse.mod4y.c, mse.mod4e.c,
                      mse.mod5y.c, mse.mod5e.c, mse.mod6y.c, mse.mod6e.c,
                      mse.mod7y.c, mse.mod7e.c, mse.mod8y.c, mse.mod8e.c,
                      lrobust.crude.y, lrobust.crude.e, lrobust.mod1y,
                      lrobust.mod1e, lrobust.mod2y, lrobust.mod2e,
                      lrobust.mod3y, lrobust.mod3e, lrobust.mod4y,
                      lrobust.mod4e, lrobust.mod5y, lrobust.mod5e,
                      lrobust.mod6y, lrobust.mod6e, lrobust.mod7y,
                      lrobust.mod7e, lrobust.mod8y, lrobust.mod8e,
                      lmse.crudecond.y, lmse.crudecond.e, lmse.mod1y.c,
                      lmse.mod1e.c, lmse.mod2y.c, lmse.mod2e.c,
                      lmse.mod3y.c, lmse.mod3e.c, lmse.mod4y.c,
                      lmse.mod4e.c, lmse.mod5y.c, lmse.mod5e.c,
                      lmse.mod6y.c, lmse.mod6e.c, lmse.mod7y.c,
                      lmse.mod7e.c, lmse.mod8y.c, lmse.mod8e.c, marg.or)
```

```
      res.frame
}

set.seed(7501345)

temp.conf <- matrix(NA, 1000, 1000)

output <- apply(temp.conf, 1, function(x)
conf.dat(1000,log(1),round(runif(1)*100000)))
conf.data<-do.call("rbind",output)



### bootstrap
### bootstrap CIs and log variance for each model (y on e)
quantile(conf.data[,1],c(0.025,0.975)) # crude y
crude.boot.y<-var(conf.data[,1])
crude.boot.y
quantile(conf.data[,3],c(0.025,0.975)) # crude e
crude.boot.e<-var(conf.data[,3])
crude.boot.e

quantile(conf.data[,5],c(0.025,0.975)) # mod 1 y
mod1.boot.y<-var(conf.data[,5])
mod1.boot.y
quantile(conf.data[,7],c(0.025,0.975)) # mod 1 e
mod1.boot.e<-var(conf.data[,7])
mod1.boot.e

quantile(conf.data[,9],c(0.025,0.975)) # mod 2 y
mod2.boot.y<-var(conf.data[,9])
mod2.boot.y
quantile(conf.data[,11],c(0.025,0.975)) # mod 2 e
mod2.boot.e<-var(conf.data[,11])
mod2.boot.e

quantile(conf.data[,13],c(0.025,0.975)) # mod 3 y
mod3.boot.y<-var(conf.data[,13])
mod3.boot.y
quantile(conf.data[,15],c(0.025,0.975)) # mod 3 e
mod3.boot.e<-var(conf.data[,15])
mod3.boot.e

quantile(conf.data[,17],c(0.025,0.975)) # mod 4 y
mod4.boot.y<-var(conf.data[,17])
mod4.boot.y
quantile(conf.data[,19],c(0.025,0.975)) # mod 4 e
mod4.boot.e<-var(conf.data[,19])
mod4.boot.e

quantile(conf.data[,21],c(0.025,0.975)) # mod 5 y
mod5.boot.y<-var(conf.data[,21])
mod5.boot.y
quantile(conf.data[,23],c(0.025,0.975)) # mod 5 e
mod5.boot.e<-var(conf.data[,23])
mod5.boot.e

quantile(conf.data[,25],c(0.025,0.975)) # mod 6 y
mod6.boot.y<-var(conf.data[,25])
```

```
mod6.boot.y
quantile(conf.data[,27],c(0.025,0.975)) # mod 6 e
mod6.boot.e<-var(conf.data[,27])
mod6.boot.e

quantile(conf.data[,29],c(0.025,0.975)) # mod 7 y
mod7.boot.y<-var(conf.data[,29])
mod7.boot.y
quantile(conf.data[,31],c(0.025,0.975)) # mod 7 e
mod7.boot.e<-var(conf.data[,31])
mod7.boot.e

quantile(conf.data[,33],c(0.025,0.975)) # mod 8 y
mod8.boot.y<-var(conf.data[,33])
mod8.boot.y
quantile(conf.data[,35],c(0.025,0.975)) # mod 8 e
mod8.boot.e<-var(conf.data[,35])
mod8.boot.e
```

Appendix B:

The R code for the Selection Bias simulations detailed in Chapter 3. The code is divided into three sections for simulations 1, 2, and 3. Each section contains two parts: the first part is the code for when the conditional odds ratio is set to 3 and the second part is the code for when the conditional odds ratio is set to 1.

```
#############################################################################
#                        Selection Bias DAG 1: Part 1                       #
#           Simulation for the Application of Prognostic Scores             #
#                              to Selection Bias                            #
#############################################################################

sim1.dat<-function(n, beta1, seed){
      set.seed(seed)
      Evar<-rbinom(n, 1, 0.3)
      Lvar<-rbinom(n, 1, 0.6)
      delta0<-log(.6/.4)
      delta1<-log(6)
      delta2<-log(5)
      probS<-exp(delta0+delta1*Evar+delta2*Lvar)/
            (1 + exp(delta0+delta1*Evar+delta2*Lvar))
      Svar<-rbinom(n,1, prob= probS)
      beta0<-log(.3/.7)
      beta2<-log(5)
      probY<-exp(beta0 + beta1*Evar + beta2*Lvar)/
            (1 + exp(beta0 + beta1*Evar + beta2*Lvar))
      Yvar<-rbinom(n, 1, prob=probY)
      dat.sim1<-data.frame(Evar, Lvar, Svar, Yvar)
      sel.sim1<-dat.sim1[dat.sim1$Svar==1,]

      # crude
       crude.sim1<-glm(Yvar ~ Evar, data=sel.sim1,
       family=binomial(link="logit"))

      # model 1
      # prognostic weights (unexposed)
      wt1.sim1 <- glm(Yvar~Lvar,data=sel.sim1[sel.sim1$Evar==0,],
                      family=binomial(link="logit"))
```

```
sel.sim1$wt1.sim1 <- predict(wt1.sim1,newdata=sel.sim1,type="response")
sel.sim1$wt1.sim1[sel.sim1$Yvar==0]<-
       1-sel.sim1$wt1.sim1[sel.sim1$Yvar==0]
progw.sim1<-1/(sel.sim1$wt1.sim1)
progsw.sim1 <- (sel.sim1$Yvar*mean(sel.sim1$Yvar) +
                   (1-sel.sim1$Yvar)*(1-mean(sel.sim1$Yvar))) * progw.sim1
mod1.sim1<-glm(Yvar ~ Evar, data=sel.sim1, weights=progsw.sim1,
               family=binomial(link="logit"))

# model 2
# full cohort DRS
wt2.sim1 <- glm(Yvar~Lvar+Evar,data=sel.sim1,family=binomial)
tmp <- sel.sim1
tmp$Evar[tmp$Evar==1] <- 0
wt2.sim1 <- predict(wt2.sim1,newdata=tmp,type="response")
wt2.sim1.s <- sel.sim1$Yvar*(mean(sel.sim1$Yvar)/wt2.sim1) +
      (1-sel.sim1$Yvar)*((1-mean(sel.sim1$Yvar))/(1-wt2.sim1))
mod2.sim1<-glm(Yvar ~ Evar, data=sel.sim1, weights=wt2.sim1.s,
               family=binomial(link="logit"))

# model 3
# modified prognostic score
wt3.sim1 <- glm(Yvar~Lvar,data=sel.sim1,family=binomial)$fitted.values
wt3.sim1.s <- sel.sim1$Yvar*(mean(sel.sim1$Yvar)/wt3.sim1) +
      (1-sel.sim1$Yvar)*((1-mean(sel.sim1$Yvar))/(1-wt3.sim1))
mod3.sim1<-glm(Yvar ~ Evar, data=sel.sim1, weights=wt3.sim1.s,
               family=binomial(link="logit"))

# model 4
# ipsw
wt4.sim1<-glm(Svar~Lvar+Evar,data=dat.sim1,family=binomial(link="logit"))
sel.sim1$wt4.sim1 <- predict(wt4.sim1, newdata=sel.sim1, type="response")
ipsw.sim1<-1/(sel.sim1$wt4.sim1)
ipsw.sim1 <-mean(dat.sim1$Svar)*ipsw.sim1
mod4.sim1<-glm(Yvar ~ Evar, data=sel.sim1, weights=ipsw.sim1,
               family=binomial(link="logit"))

# model 5
# direct adjustment
mod5.sim1<-glm(Yvar~Evar+Lvar,data=sel.sim1,
 family=binomial(link="logit"))

# model 5
# full population (true data)
mod6.sim1<-glm(Yvar ~ Evar, data=dat.sim1,family=binomial(link="logit"))

# marginal OR
pbar0 <- (beta0+beta2*Lvar)
pbar0 <- mean(1/(1+exp(-pbar0)))
pbar1 <- (beta0+beta1+beta2*Lvar)
pbar1 <- mean(1/(1+exp(-pbar1)))
marg.or <- (pbar1/(1-pbar1)) / (pbar0/(1-pbar0))
beta.crude<-summary(crude.sim1)$coef[2,1]
var.crude<-(summary(crude.sim1)$coef[2,2])^2
beta.mod1<-summary(mod1.sim1)$coef[2,1]
var.mod1<-(summary(mod1.sim1)$coef[2,2])^2
beta.mod2<-summary(mod2.sim1)$coef[2,1]
var.mod2<-(summary(mod2.sim1)$coef[2,2])^2
beta.mod3<-summary(mod3.sim1)$coef[2,1]
var.mod3<-(summary(mod3.sim1)$coef[2,2])^2
```

```
beta.mod4<-summary(mod4.sim1)$coef[2,1]
var.mod4<-(summary(mod4.sim1)$coef[2,2])^2
beta.mod5<-summary(mod5.sim1)$coef[2,1]
var.mod5<-(summary(mod5.sim1)$coef[2,2])^2
beta.mod6<-summary(mod6.sim1)$coef[2,1]
var.mod6<-(summary(mod6.sim1)$coef[2,2])^2

## insert calculation for % bias
bias.crudecond=((beta.crude-log(3))/log(3))*100
bias.crudemarg=((beta.crude-log(marg.or))/log(marg.or))*100
bias.mod1c=((beta.mod1-log(3))/log(3))*100
bias.mod1m=((beta.mod1-log(marg.or))/log(marg.or))*100
bias.mod2c=((beta.mod2-log(3))/log(3))*100
bias.mod2m=((beta.mod2-log(marg.or))/log(marg.or))*100
bias.mod3c=((beta.mod3-log(3))/log(3))*100
bias.mod3m=((beta.mod3-log(marg.or))/log(marg.or))*100
bias.mod4c=((beta.mod4-log(3))/log(3))*100
bias.mod4m=((beta.mod4-log(marg.or))/log(marg.or))*100
bias.mod5c=((beta.mod5-log(3))/log(3))*100
bias.mod5m=((beta.mod5-log(marg.or))/log(marg.or))*100
bias.mod6c=((beta.mod6-log(3))/log(3))*100
bias.mod6m=((beta.mod6-log(marg.or))/log(marg.or))*100

## robust regression variance
library(sandwich)
robust.crude <- diag(sandwich(crude.sim1))
robust.crude <- robust.crude[2]
robust.mod1 <- diag(sandwich(mod1.sim1))
robust.mod1 <-robust.mod1[2]
robust.mod2 <- diag(sandwich(mod2.sim1))
robust.mod2 <-robust.mod2[2]
robust.mod3 <- diag(sandwich(mod3.sim1))
robust.mod3 <-robust.mod3[2]
robust.mod4 <- diag(sandwich(mod4.sim1))
robust.mod4 <-robust.mod4[2]
robust.mod5 <- diag(sandwich(mod5.sim1))
robust.mod5 <-robust.mod5[2]
robust.mod6 <- diag(sandwich(mod6.sim1))
robust.mod6 <-robust.mod6[2]

## mse calculation using robust variance
mse.crudecond=((bias.crudecond/100)^2)+robust.crude
mse.crudemarg=((bias.crudemarg/100)^2)+robust.crude
mse.mod1c=((bias.mod1c/100)^2)+robust.mod1
mse.mod1m=((bias.mod1m/100)^2)+robust.mod1
mse.mod2c=((bias.mod2c/100)^2)+robust.mod2
mse.mod2m=((bias.mod2m/100)^2)+robust.mod2
mse.mod3c=((bias.mod3c/100)^2)+robust.mod3
mse.mod3m=((bias.mod3m/100)^2)+robust.mod3
mse.mod4c=((bias.mod4c/100)^2)+robust.mod4
mse.mod4m=((bias.mod4m/100)^2)+robust.mod4
mse.mod5c=((bias.mod5c/100)^2)+robust.mod5
mse.mod5m=((bias.mod5m/100)^2)+robust.mod5
mse.mod6c=((bias.mod6c/100)^2)+robust.mod6
mse.mod6m=((bias.mod6m/100)^2)+robust.mod6

## log mse calculation using robust variance
lmse.crudecond=log(((bias.crudecond/100)^2)+robust.crude)
lmse.crudemarg=log(((bias.crudemarg/100)^2)+robust.crude)
lmse.mod1c=log(((bias.mod1c/100)^2)+robust.mod1)
```

```
      lmse.mod1m=log(((bias.mod1m/100)^2)+robust.mod1)
      lmse.mod2c=log(((bias.mod2c/100)^2)+robust.mod2)
      lmse.mod2m=log(((bias.mod2m/100)^2)+robust.mod2)
      lmse.mod3c=log(((bias.mod3c/100)^2)+robust.mod3)
      lmse.mod3m=log(((bias.mod3m/100)^2)+robust.mod3)
      lmse.mod4c=log(((bias.mod4c/100)^2)+robust.mod4)
      lmse.mod4m=log(((bias.mod4m/100)^2)+robust.mod4)
      lmse.mod5c=log(((bias.mod5c/100)^2)+robust.mod5)
      lmse.mod5m=log(((bias.mod5m/100)^2)+robust.mod5)
      lmse.mod6c=log(((bias.mod6c/100)^2)+robust.mod6)
      lmse.mod6m=log(((bias.mod6m/100)^2)+robust.mod6)

      # convert to log robust variance

      lrobust.crude <- log(robust.crude)
      lrobust.mod1 <-log(robust.mod1)
      lrobust.mod2 <-log(robust.mod2)
      lrobust.mod3 <-log(robust.mod3)
      lrobust.mod4 <-log(robust.mod4)
      lrobust.mod5 <-log(robust.mod5)
      lrobust.mod6 <-log(robust.mod6)

      res.frame<-data.frame(beta.crude, var.crude, beta.mod1, var.mod1,
                            beta.mod2, var.mod2, beta.mod3, var.mod3,
                            beta.mod4, var.mod4, beta.mod5, var.mod5,
                            beta.mod6, var.mod6, bias.crudecond, bias.crudemarg,
                            bias.mod1c, bias.mod1m, bias.mod2c, bias.mod2m,
                            bias.mod3c, bias.mod3m, bias.mod4c, bias.mod4m,
                            bias.mod5c, bias.mod5m, bias.mod6c, bias.mod6m,
                            mse.crudecond, mse.crudemarg, mse.mod1c,
                            mse.mod1m, mse.mod2c, mse.mod2m, mse.mod3c,
                            mse.mod3m, mse.mod4c, mse.mod4m, mse.mod5c,
                            mse.mod5m, mse.mod6c, mse.mod6m, robust.crude,
                            robust.mod1, robust.mod2, robust.mod3, robust.mod4,
                            robust.mod5, robust.mod6, lmse.crudecond,
                            lmse.crudemarg, lmse.mod1c, lmse.mod1m,
                            lmse.mod2c, lmse.mod2m, lmse.mod3c,
                            lmse.mod3m, lmse.mod4c, lmse.mod4m, lmse.mod5c,
                            lmse.mod5m, lmse.mod6c, lmse.mod6m, lrobust.crude,
                            lrobust.mod1, lrobust.mod2, lrobust.mod3,
                            lrobust.mod4, lrobust.mod5, lrobust.mod6, marg.or)
      res.frame
}

set.seed(571031)
temp.mat <- matrix(NA, 1000, 1000)
output <- apply(temp.mat, 1, function(x)
sim1.dat(1000,log(3),round(runif(1)*100000)))
sim1data<-do.call("rbind",output)


### bootstrap CIs and log variance for each model
quantile(sim1data[,1],c(0.025,0.975)) # crude
crude.boot<-var(sim1data[,1])
crude.boot

quantile(sim1data[,3],c(0.025,0.975)) # mod 1
mod1.boot<-var(sim1data[,3])
mod1.boot
```

```
quantile(sim1data[,5],c(0.025,0.975)) # mod 2
mod2.boot<-var(sim1data[,5])
mod2.boot

quantile(sim1data[,7],c(0.025,0.975)) # mod 3
mod3.boot<-var(sim1data[,7])
mod3.boot

quantile(sim1data[,9],c(0.025,0.975)) # mod 4
mod4.boot<-var(sim1data[,9])
mod4.boot

quantile(sim1data[,11],c(0.025,0.975)) # mod 5
mod5.boot<-var(sim1data[,11])
mod5.boot

quantile(sim1data[,13],c(0.025,0.975)) # mod 6
mod6.boot<-var(sim1data[,13])
mod6.boot


##############################################################################
#                         Selection Bias DAG 1: Part 1                       #
#           Simulation for the Application of Prognostic Scores              #
#                               to Selection Bias                            #
##############################################################################

sim1.dat<-function(n, beta1, seed){
      set.seed(seed)
      Evar<-rbinom(n, 1, 0.3)
      Lvar<-rbinom(n, 1, 0.6)
      delta0<-log(.6/.4)
      delta1<-log(6)
      delta2<-log(5)
      probS<-exp(delta0+delta1*Evar+delta2*Lvar)/
            (1 + exp(delta0+delta1*Evar+delta2*Lvar))
      Svar<-rbinom(n,1, prob= probS)
      beta0<-log(.3/.7)
      beta2<-log(5)
      probY<-exp(beta0 + beta1*Evar + beta2*Lvar)/
            (1 + exp(beta0 + beta1*Evar + beta2*Lvar))
      Yvar<-rbinom(n, 1, prob=probY)
      dat.sim1<-data.frame(Evar, Lvar, Svar, Yvar)
      sel.sim1<-dat.sim1[dat.sim1$Svar==1,]

      # crude
      crude.sim1<-glm(Yvar ~ Evar, data=sel.sim1,
                      family=binomial(link="logit"))

      # model 1
      # prognostic weights (unexposed)
      wt1.sim1 <- glm(Yvar~Lvar,data=sel.sim1[sel.sim1$Evar==0,],
                      family=binomial(link="logit"))
      sel.sim1$wt1.sim1 <- predict(wt1.sim1,newdata=sel.sim1,type="response")
      sel.sim1$wt1.sim1[sel.sim1$Yvar==0]<-
       1-sel.sim1$wt1.sim1[sel.sim1$Yvar==0]
      progw.sim1<-1/(sel.sim1$wt1.sim1)
      progsw.sim1 <- (sel.sim1$Yvar*mean(sel.sim1$Yvar) +
                        (1-sel.sim1$Yvar)*(1-mean(sel.sim1$Yvar))) * progw.sim1
      mod1.sim1<-glm(Yvar ~ Evar, data=sel.sim1, weights=progsw.sim1,
```

```
               family=binomial(link="logit"))

# model 2
# full cohort DRS
wt2.sim1 <- glm(Yvar~Lvar+Evar,data=sel.sim1,family=binomial)
tmp <- sel.sim1
tmp$Evar[tmp$Evar==1] <- 0
wt2.sim1 <- predict(wt2.sim1,newdata=tmp,type="response")
wt2.sim1.s <- sel.sim1$Yvar*(mean(sel.sim1$Yvar)/wt2.sim1) +
      (1-sel.sim1$Yvar)*((1-mean(sel.sim1$Yvar))/(1-wt2.sim1))
mod2.sim1<-glm(Yvar ~ Evar, data=sel.sim1, weights=wt2.sim1.s,
               family=binomial(link="logit"))

# model 3
# modified prognostic score
wt3.sim1 <- glm(Yvar~Lvar,data=sel.sim1,family=binomial)$fitted.values
wt3.sim1.s <- sel.sim1$Yvar*(mean(sel.sim1$Yvar)/wt3.sim1) +
      (1-sel.sim1$Yvar)*((1-mean(sel.sim1$Yvar))/(1-wt3.sim1))
mod3.sim1<-glm(Yvar ~ Evar, data=sel.sim1, weights=wt3.sim1.s,
               family=binomial(link="logit"))

# model 4
# ipsw
wt4.sim1<-glm(Svar~Lvar+Evar,data=dat.sim1,family=binomial(link="logit"))
sel.sim1$wt4.sim1 <- predict(wt4.sim1, newdata=sel.sim1, type="response")
ipsw.sim1<-1/(sel.sim1$wt4.sim1)
ipsw.sim1 <-mean(dat.sim1$Svar)*ipsw.sim1
mod4.sim1<-glm(Yvar ~ Evar, data=sel.sim1, weights=ipsw.sim1,
               family=binomial(link="logit"))

# model 5
# direct adjustment
mod5.sim1<-glm(Yvar~Evar+Lvar,data=sel.sim1,
 family=binomial(link="logit"))

# model 5
# full population (true data)
mod6.sim1<-glm(Yvar ~ Evar, data=dat.sim1,family=binomial(link="logit"))


# marginal OR
pbar0 <- (beta0+beta2*Lvar)
pbar0 <- mean(1/(1+exp(-pbar0)))
pbar1 <- (beta0+beta1+beta2*Lvar)
pbar1 <- mean(1/(1+exp(-pbar1)))
marg.or <- (pbar1/(1-pbar1)) / (pbar0/(1-pbar0))
beta.crude<-summary(crude.sim1)$coef[2,1]
var.crude<-(summary(crude.sim1)$coef[2,2])^2
beta.mod1<-summary(mod1.sim1)$coef[2,1]
var.mod1<-(summary(mod1.sim1)$coef[2,2])^2
beta.mod2<-summary(mod2.sim1)$coef[2,1]
var.mod2<-(summary(mod2.sim1)$coef[2,2])^2
beta.mod3<-summary(mod3.sim1)$coef[2,1]
var.mod3<-(summary(mod3.sim1)$coef[2,2])^2
beta.mod4<-summary(mod4.sim1)$coef[2,1]
var.mod4<-(summary(mod4.sim1)$coef[2,2])^2
beta.mod5<-summary(mod5.sim1)$coef[2,1]
var.mod5<-(summary(mod5.sim1)$coef[2,2])^2
beta.mod6<-summary(mod6.sim1)$coef[2,1]
var.mod6<-(summary(mod6.sim1)$coef[2,2])^2
```

```
      ## robust regression variance
      library(sandwich)
      robust.crude <- diag(sandwich(crude.sim1))
      robust.crude <- robust.crude[2]
      robust.mod1 <- diag(sandwich(mod1.sim1))
      robust.mod1 <-robust.mod1[2]
      robust.mod2 <- diag(sandwich(mod2.sim1))
      robust.mod2 <-robust.mod2[2]
      robust.mod3 <- diag(sandwich(mod3.sim1))
      robust.mod3 <-robust.mod3[2]
      robust.mod4 <- diag(sandwich(mod4.sim1))
      robust.mod4 <-robust.mod4[2]
      robust.mod5 <- diag(sandwich(mod5.sim1))
      robust.mod5 <-robust.mod5[2]
      robust.mod6 <- diag(sandwich(mod6.sim1))
      robust.mod6 <-robust.mod6[2]

      ## mse calculation using robust variance
      mse.crudecond=((beta.crude)^2)+robust.crude
      mse.mod1c=((beta.mod1)^2)+robust.mod1
      mse.mod2c=((beta.mod2)^2)+robust.mod2
      mse.mod3c=((beta.mod3)^2)+robust.mod3
      mse.mod4c=((beta.mod4)^2)+robust.mod4
      mse.mod5c=((beta.mod5)^2)+robust.mod5
      mse.mod6c=((beta.mod6)^2)+robust.mod6

      ## log mse calculation using robust variance
      lmse.crudecond=log(((beta.crude)^2)+robust.crude)
      lmse.mod1c=log(((beta.mod1)^2)+robust.mod1)
      lmse.mod2c=log(((beta.mod2)^2)+robust.mod2)
      lmse.mod3c=log(((beta.mod3)^2)+robust.mod3)
      lmse.mod4c=log(((beta.mod4)^2)+robust.mod4)
      lmse.mod5c=log(((beta.mod5)^2)+robust.mod5)
      lmse.mod6c=log(((beta.mod6)^2)+robust.mod6)

      # convert to log robust variance

      lrobust.crude <- log(robust.crude)
      lrobust.mod1 <-log(robust.mod1)
      lrobust.mod2 <-log(robust.mod2)
      lrobust.mod3 <-log(robust.mod3)
      lrobust.mod4 <-log(robust.mod4)
      lrobust.mod5 <-log(robust.mod5)
      lrobust.mod6 <-log(robust.mod6)

      res.frame<-data.frame(beta.crude, var.crude, beta.mod1, var.mod1,
                            beta.mod2, var.mod2, beta.mod3, var.mod3,
                            beta.mod4, var.mod4, beta.mod5, var.mod5,
                            beta.mod6, var.mod6, mse.crudecond, mse.mod1c,
                            mse.mod2c, mse.mod3c, mse.mod3c, mse.mod4c,
                            mse.mod5c, mse.mod6c, robust.crude, robust.mod1,
                            robust.mod2, robust.mod3, robust.mod4, robust.mod5,
                            robust.mod6, lmse.crudecond, lmse.mod1c,
                            lmse.mod2c, lmse.mod3c, lmse.mod3c, lmse.mod4c,
                            lmse.mod5c, lmse.mod6c, lrobust.crude,lrobust.mod1,
                            lrobust.mod2, lrobust.mod3, lrobust.mod4,
                            lrobust.mod5, lrobust.mod6, marg.or)
      res.frame
}
```

```
set.seed(571031)
temp.mat <- matrix(NA, 1000, 1000)
output <- apply(temp.mat, 1, function(x)
sim1.dat(1000,log(1),round(runif(1)*100000)))
sim1data<-do.call("rbind",output)

### bootstrap CIs and log variance for each model
quantile(sim1data[,1],c(0.025,0.975)) # crude
crude.boot<-var(sim1data[,1])
crude.boot

quantile(sim1data[,3],c(0.025,0.975)) # mod 1
mod1.boot<-var(sim1data[,3])
mod1.boot

quantile(sim1data[,5],c(0.025,0.975)) # mod 2
mod2.boot<-var(sim1data[,5])
mod2.boot

quantile(sim1data[,7],c(0.025,0.975)) # mod 3
mod3.boot<-var(sim1data[,7])
mod3.boot

quantile(sim1data[,9],c(0.025,0.975)) # mod 4
mod4.boot<-var(sim1data[,9])
mod4.boot

quantile(sim1data[,11],c(0.025,0.975)) # mod 5
mod5.boot<-var(sim1data[,11])
mod5.boot

quantile(sim1data[,13],c(0.025,0.975)) # mod 6
mod6.boot<-var(sim1data[,13])
mod6.boot

#############################################################################
#                        Selection Bias DAG 2: Part 1                       #
#           Simulation for the Application of Prognostic Scores             #
#                               to Selection Bias                           #
#############################################################################

sim2.dat <- function(n,gamma3,seed){
      set.seed(seed)
      beta0<-log(.6/.4)
      beta1<-log(1/3)
      gamma0<-log(.3/.7)
      gamma1<-log(2)
      gamma2<-log(5)
      delta0<-log(.6/.4)
      delta1<-log(6)
      delta2<-log(5)
      Cvar<-rbinom(n, 1, 0.3)
      Lvar<-rbinom(n, 1, 0.6)
      probE<-exp(beta0+beta1 * Cvar)/(1 + exp(beta0+beta1*Cvar))
      Evar<-rbinom(n, 1, prob=probE)
      probY<-exp(gamma0+gamma1*Cvar+gamma2*Lvar+gamma3*Evar)/
            (1 + exp(gamma0+gamma1*Cvar+gamma2*Lvar+gamma3*Evar))
      Yvar<-rbinom(n, 1, prob=probY)
      probS<-exp(delta0+delta1*Evar+delta2*Lvar)/
```

```
      (1 + exp(delta0+delta1*Evar+delta2*Lvar))
Svar<-rbinom(n,1, prob= probS)
dat<-data.frame(Evar,Yvar,Svar,Lvar,Cvar)
sel<-dat[dat$Svar==1,]


# prognostic weights-LC
wt1 <- glm(Yvar~Lvar + Cvar,data=sel[sel$Evar==0,],
           family=binomial(link="logit"))
sel$wt1 <- predict(wt1,newdata=sel,type="response")
sel$wt1[sel$Yvar==0] <- 1-sel$wt1[sel$Yvar==0]
prog1w<-1/(sel$wt1)
prog1sw <- (sel$Yvar*mean(sel$Yvar) +
                  (1-sel$Yvar)*(1-mean(sel$Yvar))) * prog1w
# prognostic weights-L
wt2 <- glm(Yvar ~ Lvar, data=sel[sel$Evar==0,],
           family=binomial(link="logit"))
sel$wt2 <- predict(wt2, newdata=sel, type="response")
sel$wt2[sel$Yvar==0] <- 1-sel$wt2[sel$Yvar==0]
prog2w<-1/(sel$wt2)
prog2sw <- (sel$Yvar*mean(sel$Yvar) +
                  (1-sel$Yvar)*(1-mean(sel$Yvar))) * prog2w

# prog weights- LC in full population
wt3 <- glm(Yvar~Lvar+Cvar+Evar,data=sel,family=binomial)
tmp <- sel
tmp$Evar[tmp$Evar==1] <- 0
wt3 <- predict(wt3,newdata=tmp,type="response")
prog.lc.full <- sel$Yvar*(mean(sel$Yvar)/wt3) +
      (1-sel$Yvar)*((1-mean(sel$Yvar))/(1-wt3))

# prog weights- L in full population
wt4 <- glm(Yvar~Lvar+Evar,data=sel,family=binomial)
wt4 <- predict(wt4,newdata=tmp,type="response")
prog.l.full <- sel$Yvar*(mean(sel$Yvar)/wt4) +
      (1-sel$Yvar)*((1-mean(sel$Yvar))/(1-wt4))

# modified prog weights- LC
wt5<- glm(Yvar~Lvar+Cvar,data=sel,family=binomial)$fitted.values
prog.lc.mod <- sel$Yvar*(mean(sel$Yvar)/wt5) +
      (1-sel$Yvar)*((1-mean(sel$Yvar))/(1-wt5))

# modified prog weights- L
wt6 <- glm(Yvar~Lvar,data=sel,family=binomial)$fitted.values
prog.l.mod <- sel$Yvar*(mean(sel$Yvar)/wt6) +
      (1-sel$Yvar)*((1-mean(sel$Yvar))/(1-wt6))

# ipw- C
wt3 <- glm(Evar~Cvar, data=sel, family=binomial(link="logit"))
sel$wt3 <- predict(wt3, newdata=sel, type="response")
sel$wt3[sel$Evar==0] <- 1-sel$wt3[sel$Evar==0]
ipew<-1/(sel$wt3)
ipesw <- (sel$Evar*mean(sel$Evar)+(1-sel$Evar)*(1-mean(sel$Evar))) * ipew
# combo weights
ComboW<-ipesw*prog2sw # prog L, ipw
ComboW.full<-ipesw*prog.l.full   # prog L full, ipw
ComboW.mod<-ipesw*prog.l.mod   # prog L modified, ipw
#
crude<-glm(Yvar~Evar, data=sel, family=binomial(link="logit")) ## crude
mod1<-glm(Yvar ~ Evar, data=sel, weights=prog1sw,
```

```
          family=binomial(link="logit")) #prognostic LC
mod2<-glm(Yvar ~ Evar, data=sel, weights=prog.lc.full,
          family=binomial(link="logit")) # prognostic full LC
mod3<-glm(Yvar ~ Evar, data=sel, weights=prog.lc.mod,
          family=binomial(link="logit")) # modified prognostic LC
mod4<-glm(Yvar ~ Evar, data=sel, weights=ComboW,
          family=binomial(link="logit")) # combo prog L, ipw
mod5<-glm(Yvar ~ Evar, data=sel, weights=ComboW.full,
          family=binomial(link="logit")) # combo prog full L, ipw
mod6<-glm(Yvar ~ Evar, data=sel, weights=ComboW.mod,
          family=binomial(link="logit")) # combo prog modified L, ipw
mod7<-glm(Yvar~ Evar+Lvar+Cvar, data=sel,
          family=binomial(link="logit")) ## direct adjust
#
# partial adjustment models
mod8<-glm(Yvar ~ Evar + Cvar, data=dat,
          family=binomial(link="logit")) # direct adjust C, no selection
mod9<-glm(Yvar ~ Evar, data=dat,
          family=binomial(link="logit"))
       ## effect of confounder on marginal
mod10<-glm(Yvar ~ Evar +Cvar, data=sel,
           family=binomial(link="logit")) ## effect of selection
mod11<-glm(Yvar~ Evar+Lvar, data=sel,
           family=binomial(link="logit"))
            ### effect of the confounder on conditional
#
#
beta.mod1<-summary(mod1)$coef[2,1]
var.mod1<-(summary(mod1)$coef[2,2])^2
beta.mod2<-summary(mod2)$coef[2,1]
var.mod2<-(summary(mod2)$coef[2,2])^2
beta.mod3<-summary(mod3)$coef[2,1]
var.mod3<-(summary(mod3)$coef[2,2])^2
beta.mod4<-summary(mod4)$coef[2,1]
var.mod4<-(summary(mod4)$coef[2,2])^2
beta.mod5<-summary(mod5)$coef[2,1]
var.mod5<-(summary(mod5)$coef[2,2])^2
beta.crude<-summary(crude)$coef[2,1]
var.crude<-(summary(crude)$coef[2,2])^2
beta.mod6<-summary(mod6)$coef[2,1]
var.mod6<-(summary(mod6)$coef[2,2])^2
beta.mod7<-summary(mod7)$coef[2,1]
var.mod7<-(summary(mod7)$coef[2,2])^2
beta.mod8<-summary(mod8)$coef[2,1]
var.mod8<-(summary(mod8)$coef[2,2])^2
beta.mod9<-summary(mod9)$coef[2,1]
var.mod9<-(summary(mod9)$coef[2,2])^2
beta.mod10<-summary(mod10)$coef[2,1]
var.mod10<-(summary(mod10)$coef[2,2])^2
beta.mod11<-summary(mod11)$coef[2,1]
var.mod11<-(summary(mod11)$coef[2,2])^2
# marginal or
pbar0 <- (gamma0+gamma1*Cvar+gamma2*Lvar)
pbar0 <- mean(1/(1+exp(-pbar0)))
pbar1 <- (gamma0+gamma1*Cvar+gamma2*Lvar+gamma3)
pbar1 <- mean(1/(1+exp(-pbar1)))
marg.or <- (pbar1/(1-pbar1)) / (pbar0/(1-pbar0))
#
## insert calculation for % bias for marg and conditional
bias.crudecond=((beta.crude-log(3))/log(3))*100
```

```
bias.crudemarg=((beta.crude-log(marg.or))/log(marg.or))*100
bias.mod1c=((beta.mod1-log(3))/log(3))*100
bias.mod1m=((beta.mod1-log(marg.or))/log(marg.or))*100
bias.mod2c=((beta.mod2-log(3))/log(3))*100
bias.mod2m=((beta.mod2-log(marg.or))/log(marg.or))*100
bias.mod3c=((beta.mod3-log(3))/log(3))*100
bias.mod3m=((beta.mod3-log(marg.or))/log(marg.or))*100
bias.mod4c=((beta.mod4-log(3))/log(3))*100
bias.mod4m=((beta.mod4-log(marg.or))/log(marg.or))*100
bias.mod5c=((beta.mod5-log(3))/log(3))*100
bias.mod5m=((beta.mod5-log(marg.or))/log(marg.or))*100
bias.mod6c=((beta.mod6-log(3))/log(3))*100
bias.mod6m=((beta.mod6-log(marg.or))/log(marg.or))*100
bias.mod7c=((beta.mod7-log(3))/log(3))*100
bias.mod7m=((beta.mod7-log(marg.or))/log(marg.or))*100
# don't need bias, mse for partial models
#
## robust regression variance
library(sandwich)
robust.crude <- diag(sandwich(crude))
robust.crude <- robust.crude[2]
robust.mod1 <- diag(sandwich(mod1))
robust.mod1 <-robust.mod1[2]
robust.mod2 <- diag(sandwich(mod2))
robust.mod2 <-robust.mod2[2]
robust.mod3 <- diag(sandwich(mod3))
robust.mod3 <-robust.mod3[2]
robust.mod4 <- diag(sandwich(mod4))
robust.mod4 <-robust.mod4[2]
robust.mod5 <- diag(sandwich(mod5))
robust.mod5 <-robust.mod5[2]
robust.mod6 <- diag(sandwich(mod6))
robust.mod6 <-robust.mod6[2]
robust.mod7 <- diag(sandwich(mod7))
robust.mod7 <-robust.mod7[2]
robust.mod8 <- diag(sandwich(mod8))
robust.mod8 <-robust.mod8[2]
robust.mod9 <- diag(sandwich(mod9))
robust.mod9 <-robust.mod9[2]
robust.mod10 <- diag(sandwich(mod10))
robust.mod10 <-robust.mod10[2]
robust.mod11 <- diag(sandwich(mod11))
robust.mod11 <-robust.mod11[2]

## mse calculation- used robust variance for the variance calculation
mse.crudecond=((bias.crudecond/100)^2)+robust.crude
mse.crudemarg=((bias.crudemarg/100)^2)+robust.crude
mse.mod1c=((bias.mod1c/100)^2)+robust.mod1
mse.mod1m=((bias.mod1m/100)^2)+robust.mod1
mse.mod2c=((bias.mod2c/100)^2)+robust.mod2
mse.mod2m=((bias.mod2m/100)^2)+robust.mod2
mse.mod3c=((bias.mod3c/100)^2)+robust.mod3
mse.mod3m=((bias.mod3m/100)^2)+robust.mod3
mse.mod4c=((bias.mod4c/100)^2)+robust.mod4
mse.mod4m=((bias.mod4m/100)^2)+robust.mod4
mse.mod5c=((bias.mod5c/100)^2)+robust.mod5
mse.mod5m=((bias.mod5m/100)^2)+robust.mod5
mse.mod6c=((bias.mod6c/100)^2)+robust.mod6
mse.mod6m=((bias.mod6m/100)^2)+robust.mod6
mse.mod7c=((bias.mod7c/100)^2)+robust.mod7
```

```
mse.mod7m=((bias.mod7m/100)^2)+robust.mod7

## log mse calculation- used robust variance for the variance calculation
lmse.crudecond=log(((bias.crudecond/100)^2)+robust.crude)
lmse.crudemarg=log(((bias.crudemarg/100)^2)+robust.crude)
lmse.mod1c=log(((bias.mod1c/100)^2)+robust.mod1)
lmse.mod1m=log(((bias.mod1m/100)^2)+robust.mod1)
lmse.mod2c=log(((bias.mod2c/100)^2)+robust.mod2)
lmse.mod2m=log(((bias.mod2m/100)^2)+robust.mod2)
lmse.mod3c=log(((bias.mod3c/100)^2)+robust.mod3)
lmse.mod3m=log(((bias.mod3m/100)^2)+robust.mod3)
lmse.mod4c=log(((bias.mod4c/100)^2)+robust.mod4)
lmse.mod4m=log(((bias.mod4m/100)^2)+robust.mod4)
lmse.mod5c=log(((bias.mod5c/100)^2)+robust.mod5)
lmse.mod5m=log(((bias.mod5m/100)^2)+robust.mod5)
lmse.mod6c=log(((bias.mod6c/100)^2)+robust.mod6)
lmse.mod6m=log(((bias.mod6m/100)^2)+robust.mod6)
lmse.mod7c=log(((bias.mod7c/100)^2)+robust.mod7)
lmse.mod7m=log(((bias.mod7m/100)^2)+robust.mod7)

# convert to log robust variance
lrobust.crude <- log(robust.crude)
lrobust.mod1 <-log(robust.mod1)
lrobust.mod2 <-log(robust.mod2)
lrobust.mod3 <-log(robust.mod3)
lrobust.mod4 <-log(robust.mod4)
lrobust.mod5 <-log(robust.mod5)
lrobust.mod6 <-log(robust.mod6)
lrobust.mod7 <-log(robust.mod7)
lrobust.mod8 <-log(robust.mod8)
lrobust.mod9 <-log(robust.mod9)
lrobust.mod10 <-log(robust.mod10)
lrobust.mod11 <-log(robust.mod11)

res.frame<-data.frame(beta.crude, var.crude, beta.mod1, var.mod1,
                      beta.mod2, var.mod2, beta.mod3, var.mod3,
                      beta.mod4, var.mod4, beta.mod5, var.mod5,
                      beta.mod6, var.mod6, beta.mod7, var.mod7,
                      beta.mod8, var.mod8, beta.mod9, var.mod9,
                      beta.mod10, var.mod10, beta.mod11, var.mod11,
                      marg.or, bias.crudecond, bias.crudemarg,bias.mod1c,
                      bias.mod1m, bias.mod2c, bias.mod2m, bias.mod3c,
                      bias.mod3m, bias.mod4c, bias.mod4m, bias.mod5c,
                      bias.mod5m, bias.mod6c, bias.mod6m, bias.mod7c,
                      bias.mod7m, robust.crude, robust.mod1, robust.mod2,
                      robust.mod3, robust.mod4, robust.mod5, robust.mod6,
                      robust.mod7, robust.mod8, robust.mod9,robust.mod10,
                      robust.mod11,mse.crudecond,mse.crudemarg,mse.mod1c,
                      mse.mod1m, mse.mod2c,mse.mod2m,mse.mod3c,mse.mod3m,
                      mse.mod4c, mse.mod4m,mse.mod5c,mse.mod5m,mse.mod6c,
                      mse.mod6m, mse.mod7c, mse.mod7m, lrobust.crude,
                      lrobust.mod1, lrobust.mod2, lrobust.mod3,
                      lrobust.mod4, lrobust.mod5, lrobust.mod6,
                      lrobust.mod7, lrobust.mod8, lrobust.mod9,
                      lrobust.mod10, lrobust.mod11, lmse.crudecond,
                      lmse.crudemarg, lmse.mod1c, lmse.mod1m, lmse.mod2c,
                      lmse.mod2m, lmse.mod3c, lmse.mod3m,
                      lmse.mod4c, lmse.mod4m, lmse.mod5c, lmse.mod5m,
                      lmse.mod6c, lmse.mod6m, lmse.mod7c, lmse.mod7m)
res.frame
```

```
}

set.seed(987523)
#### review the issue with the seed
temp.mat2 <- matrix(NA, 1000, 1000)

output <- apply(temp.mat2, 1, function(x)
sim2.dat(1000,log(3),round(runif(1)*100000)))
sim2data<-do.call("rbind",output)

### bootstrap CIs and log variance for each model
quantile(sim2data[,1],c(0.025,0.975)) # crude
crude.boot<-var(sim2data[,1])
crude.boot
quantile(sim2data[,3],c(0.025,0.975)) # mod 1
mod1.boot<-var(sim2data[,3])
mod1.boot
quantile(sim2data[,5],c(0.025,0.975)) # mod 2
mod2.boot<-var(sim2data[,5])
mod2.boot
quantile(sim2data[,7],c(0.025,0.975)) # mod 3
mod3.boot<-var(sim2data[,7])
mod3.boot
quantile(sim2data[,9],c(0.025,0.975)) # mod 4
mod4.boot<-var(sim2data[,9])
mod4.boot
quantile(sim2data[,11],c(0.025,0.975)) # mod 5
mod5.boot<-var(sim2data[,11])
mod5.boot
quantile(sim2data[,13],c(0.025,0.975)) # mod 6
mod6.boot<-var(sim2data[,13])
mod6.boot
quantile(sim2data[,15],c(0.025,0.975)) # mod 7
mod7.boot<-var(sim2data[,15])
mod7.boot
quantile(sim2data[,17],c(0.025,0.975)) # mod 8
mod8.boot<-var(sim2data[,17])
mod8.boot
quantile(sim2data[,19],c(0.025,0.975)) # mod 9
mod9.boot<-var(sim2data[,19])
mod9.boot
quantile(sim2data[,21],c(0.025,0.975)) # mod 10
mod10.boot<-var(sim2data[,21])
mod10.boot
quantile(sim2data[,23],c(0.025,0.975)) # mod 11
mod11.boot<-var(sim2data[,23])
mod11.boot

##############################################################################
#                          Selection Bias DAG 2: Part 2                      #
#           Simulation for the Application of Prognostic Scores              #
#                                to Selection Bias                           #
##############################################################################

sim2.dat <- function(n,gamma3,seed){
      set.seed(seed)
      beta0<-log(.6/.4)
      beta1<-log(1/3)
      gamma0<-log(.3/.7)
      gamma1<-log(2)
```

```
gamma2<-log(5)
delta0<-log(.6/.4)
delta1<-log(6)
delta2<-log(5)
Cvar<-rbinom(n, 1, 0.3)
Lvar<-rbinom(n, 1, 0.6)
probE<-exp(beta0+beta1 * Cvar)/(1 + exp(beta0+beta1*Cvar))
Evar<-rbinom(n, 1, prob=probE)
probY<-exp(gamma0+gamma1*Cvar+gamma2*Lvar+gamma3*Evar)/
      (1 + exp(gamma0+gamma1*Cvar+gamma2*Lvar+gamma3*Evar))
Yvar<-rbinom(n, 1, prob=probY)
probS<-exp(delta0+delta1*Evar+delta2*Lvar)/
      (1 + exp(delta0+delta1*Evar+delta2*Lvar))
Svar<-rbinom(n,1, prob= probS)
dat<-data.frame(Evar,Yvar,Svar,Lvar,Cvar)
sel<-dat[dat$Svar==1,]
# prognostic weights-LC
wt1 <- glm(Yvar~Lvar + Cvar,data=sel[sel$Evar==0,],
           family=binomial(link="logit"))
sel$wt1 <- predict(wt1,newdata=sel,type="response")
sel$wt1[sel$Yvar==0] <- 1-sel$wt1[sel$Yvar==0]
prog1w<-1/(sel$wt1)
prog1sw <- (sel$Yvar*mean(sel$Yvar) +
                  (1-sel$Yvar)*(1-mean(sel$Yvar))) * prog1w
# prognostic weights-L
wt2 <- glm(Yvar ~ Lvar, data=sel[sel$Evar==0,],
           family=binomial(link="logit"))
sel$wt2 <- predict(wt2, newdata=sel, type="response")
sel$wt2[sel$Yvar==0] <- 1-sel$wt2[sel$Yvar==0]
prog2w<-1/(sel$wt2)
prog2sw <- (sel$Yvar*mean(sel$Yvar) +
                  (1-sel$Yvar)*(1-mean(sel$Yvar))) * prog2w

# prog weights- LC in full population
wt3 <- glm(Yvar~Lvar+Cvar+Evar,data=sel,family=binomial)
tmp <- sel
tmp$Evar[tmp$Evar==1] <- 0
wt3 <- predict(wt3,newdata=tmp,type="response")
prog.lc.full <- sel$Yvar*(mean(sel$Yvar)/wt3) +
      (1-sel$Yvar)*((1-mean(sel$Yvar))/(1-wt3))

# prog weights- L in full population
wt4 <- glm(Yvar~Lvar+Evar,data=sel,family=binomial)
wt4 <- predict(wt4,newdata=tmp,type="response")
prog.l.full <- sel$Yvar*(mean(sel$Yvar)/wt4) +
      (1-sel$Yvar)*((1-mean(sel$Yvar))/(1-wt4))

# modified prog weights- LC
wt5<- glm(Yvar~Lvar+Cvar,data=sel,family=binomial)$fitted.values
prog.lc.mod <- sel$Yvar*(mean(sel$Yvar)/wt5) +
      (1-sel$Yvar)*((1-mean(sel$Yvar))/(1-wt5))

# modified prog weights- L
wt6 <- glm(Yvar~Lvar,data=sel,family=binomial)$fitted.values
prog.l.mod <- sel$Yvar*(mean(sel$Yvar)/wt6) +
      (1-sel$Yvar)*((1-mean(sel$Yvar))/(1-wt6))

# ipw- C
wt3 <- glm(Evar~Cvar, data=sel, family=binomial(link="logit"))
sel$wt3 <- predict(wt3, newdata=sel, type="response")
```

```
sel$wt3[sel$Evar==0] <- 1-sel$wt3[sel$Evar==0]
ipew<-1/(sel$wt3)
ipesw <- (sel$Evar*mean(sel$Evar) +
               (1-sel$Evar)*(1-mean(sel$Evar))) * ipew
# combo weights
ComboW<-ipesw*prog2sw # prog L, ipw
ComboW.full<-ipesw*prog.l.full   # prog L full, ipw
ComboW.mod<-ipesw*prog.l.mod   # prog L modified, ipw
#
crude<-glm(Yvar~Evar, data=sel, family=binomial(link="logit")) ## crude
mod1<-glm(Yvar ~ Evar, data=sel, weights=prog1sw,
          family=binomial(link="logit")) #prognostic LC
mod2<-glm(Yvar ~ Evar, data=sel, weights=prog.lc.full,
          family=binomial(link="logit")) # prognostic full LC
mod3<-glm(Yvar ~ Evar, data=sel, weights=prog.lc.mod,
          family=binomial(link="logit")) # modified prognostic LC
mod4<-glm(Yvar ~ Evar, data=sel, weights=ComboW,
          family=binomial(link="logit")) # combo prog L, ipw
mod5<-glm(Yvar ~ Evar, data=sel, weights=ComboW.full,
          family=binomial(link="logit")) # combo prog full L, ipw
mod6<-glm(Yvar ~ Evar, data=sel, weights=ComboW.mod,
          family=binomial(link="logit")) # combo prog modified L, ipw
mod7<-glm(Yvar~ Evar+Lvar+Cvar, data=sel,
          family=binomial(link="logit")) ## direct adjust
#
# partial adjustment models
mod8<-glm(Yvar ~ Evar + Cvar, data=dat,
          family=binomial(link="logit")) # direct adjust C, no selection
mod9<-glm(Yvar ~ Evar, data=dat,
          family=binomial(link="logit"))
       ## effect of confounder on marginal
mod10<-glm(Yvar ~ Evar +Cvar, data=sel,
           family=binomial(link="logit")) ## effect of selection
mod11<-glm(Yvar~ Evar+Lvar, data=sel,
           family=binomial(link="logit"))
            ### effect of the confounder on conditional
#
#
beta.mod1<-summary(mod1)$coef[2,1]
var.mod1<-(summary(mod1)$coef[2,2])^2
beta.mod2<-summary(mod2)$coef[2,1]
var.mod2<-(summary(mod2)$coef[2,2])^2
beta.mod3<-summary(mod3)$coef[2,1]
var.mod3<-(summary(mod3)$coef[2,2])^2
beta.mod4<-summary(mod4)$coef[2,1]
var.mod4<-(summary(mod4)$coef[2,2])^2
beta.mod5<-summary(mod5)$coef[2,1]
var.mod5<-(summary(mod5)$coef[2,2])^2
beta.crude<-summary(crude)$coef[2,1]
var.crude<-(summary(crude)$coef[2,2])^2
beta.mod6<-summary(mod6)$coef[2,1]
var.mod6<-(summary(mod6)$coef[2,2])^2
beta.mod7<-summary(mod7)$coef[2,1]
var.mod7<-(summary(mod7)$coef[2,2])^2
beta.mod8<-summary(mod8)$coef[2,1]
var.mod8<-(summary(mod8)$coef[2,2])^2
beta.mod9<-summary(mod9)$coef[2,1]
var.mod9<-(summary(mod9)$coef[2,2])^2
beta.mod10<-summary(mod10)$coef[2,1]
var.mod10<-(summary(mod10)$coef[2,2])^2
```

```
beta.mod11<-summary(mod11)$coef[2,1]
var.mod11<-(summary(mod11)$coef[2,2])^2
# marginal or
pbar0 <- (gamma0+gamma1*Cvar+gamma2*Lvar)
pbar0 <- mean(1/(1+exp(-pbar0)))
pbar1 <- (gamma0+gamma1*Cvar+gamma2*Lvar+gamma3)
pbar1 <- mean(1/(1+exp(-pbar1)))
marg.or <- (pbar1/(1-pbar1)) / (pbar0/(1-pbar0))

# don't need mse for partial models
#
## robust regression variance
library(sandwich)
robust.crude <- diag(sandwich(crude))
robust.crude <- robust.crude[2]
robust.mod1 <- diag(sandwich(mod1))
robust.mod1 <-robust.mod1[2]
robust.mod2 <- diag(sandwich(mod2))
robust.mod2 <-robust.mod2[2]
robust.mod3 <- diag(sandwich(mod3))
robust.mod3 <-robust.mod3[2]
robust.mod4 <- diag(sandwich(mod4))
robust.mod4 <-robust.mod4[2]
robust.mod5 <- diag(sandwich(mod5))
robust.mod5 <-robust.mod5[2]
robust.mod6 <- diag(sandwich(mod6))
robust.mod6 <-robust.mod6[2]
robust.mod7 <- diag(sandwich(mod7))
robust.mod7 <-robust.mod7[2]
robust.mod8 <- diag(sandwich(mod8))
robust.mod8 <-robust.mod8[2]
robust.mod9 <- diag(sandwich(mod9))
robust.mod9 <-robust.mod9[2]
robust.mod10 <- diag(sandwich(mod10))
robust.mod10 <-robust.mod10[2]
robust.mod11 <- diag(sandwich(mod11))
robust.mod11 <-robust.mod11[2]

## mse calculation- used robust variance for the variance calculation
mse.crudecond=((beta.crude)^2)+robust.crude
mse.mod1c=((beta.mod1)^2)+robust.mod1
mse.mod2c=((beta.mod2)^2)+robust.mod2
mse.mod3c=((beta.mod3)^2)+robust.mod3
mse.mod4c=((beta.mod4)^2)+robust.mod4
mse.mod5c=((beta.mod5)^2)+robust.mod5
mse.mod6c=((beta.mod6)^2)+robust.mod6
mse.mod7c=((beta.mod7)^2)+robust.mod7

## log mse calculation- used robust variance for the variance calculation
lmse.crudecond=log(((beta.crude)^2)+robust.crude)
lmse.mod1c=log(((beta.mod1)^2)+robust.mod1)
lmse.mod2c=log(((beta.mod2)^2)+robust.mod2)
lmse.mod3c=log(((beta.mod3)^2)+robust.mod3)
lmse.mod4c=log(((beta.mod4)^2)+robust.mod4)
lmse.mod5c=log(((beta.mod5)^2)+robust.mod5)
lmse.mod6c=log(((beta.mod6)^2)+robust.mod6)
lmse.mod7c=log(((beta.mod7)^2)+robust.mod7)

# convert to log robust variance
lrobust.crude <- log(robust.crude)
```

```
      lrobust.mod1 <-log(robust.mod1)
      lrobust.mod2 <-log(robust.mod2)
      lrobust.mod3 <-log(robust.mod3)
      lrobust.mod4 <-log(robust.mod4)
      lrobust.mod5 <-log(robust.mod5)
      lrobust.mod6 <-log(robust.mod6)
      lrobust.mod7 <-log(robust.mod7)
      lrobust.mod8 <-log(robust.mod8)
      lrobust.mod9 <-log(robust.mod9)
      lrobust.mod10 <-log(robust.mod10)
      lrobust.mod11 <-log(robust.mod11)

      res.frame<-data.frame(beta.crude, var.crude, beta.mod1, var.mod1,
                            beta.mod2, var.mod2, beta.mod3, var.mod3,
                            beta.mod4, var.mod4, beta.mod5, var.mod5,
                            beta.mod6, var.mod6, beta.mod7, var.mod7,
                            beta.mod8, var.mod8, beta.mod9, var.mod9,
                            beta.mod10, var.mod10, beta.mod11, var.mod11,
                            marg.or, robust.crude, robust.mod1, robust.mod2,
                            robust.mod3, robust.mod4, robust.mod5, robust.mod6,
                            robust.mod7, robust.mod8, robust.mod9,robust.mod10,
                            robust.mod11, mse.crudecond, mse.mod1c, mse.mod2c,
                            mse.mod3c, mse.mod4c, mse.mod5c, mse.mod6c,
                            mse.mod7c, lrobust.crude, lrobust.mod1,
                            lrobust.mod2, lrobust.mod3, lrobust.mod4,
                            lrobust.mod5, lrobust.mod6, lrobust.mod7,
                            lrobust.mod8, lrobust.mod9, lrobust.mod10,
                            lrobust.mod11, lmse.crudecond, lmse.mod1c,
                            lmse.mod2c, lmse.mod3c, lmse.mod4c,
                            lmse.mod5c, lmse.mod6c, lmse.mod7c)
      res.frame
}

set.seed(987523)
#### review the issue with the seed
temp.mat2 <- matrix(NA, 1000, 1000)

output <- apply(temp.mat2, 1, function(x)
sim2.dat(1000,log(1),round(runif(1)*100000)))
sim2data<-do.call("rbind",output)

### bootstrap CIs and log variance for each model
quantile(sim2data[,1],c(0.025,0.975)) # crude
crude.boot<-var(sim2data[,1])
crude.boot
quantile(sim2data[,3],c(0.025,0.975)) # mod 1
mod1.boot<-var(sim2data[,3])
mod1.boot
quantile(sim2data[,5],c(0.025,0.975)) # mod 2
mod2.boot<-var(sim2data[,5])
mod2.boot
quantile(sim2data[,7],c(0.025,0.975)) # mod 3
mod3.boot<-var(sim2data[,7])
mod3.boot
quantile(sim2data[,9],c(0.025,0.975)) # mod 4
mod4.boot<-var(sim2data[,9])
mod4.boot
quantile(sim2data[,11],c(0.025,0.975)) # mod 5
mod5.boot<-var(sim2data[,11])
mod5.boot
```

```
quantile(sim2data[,13],c(0.025,0.975)) # mod 6
mod6.boot<-var(sim2data[,13])
mod6.boot
quantile(sim2data[,15],c(0.025,0.975)) # mod 7
mod7.boot<-var(sim2data[,15])
mod7.boot
quantile(sim2data[,17],c(0.025,0.975)) # mod 8
mod8.boot<-var(sim2data[,17])
mod8.boot
quantile(sim2data[,19],c(0.025,0.975)) # mod 9
mod9.boot<-var(sim2data[,19])
mod9.boot
quantile(sim2data[,21],c(0.025,0.975)) # mod 10
mod10.boot<-var(sim2data[,21])
mod10.boot
quantile(sim2data[,23],c(0.025,0.975)) # mod 11
mod11.boot<-var(sim2data[,23])
mod11.boot

##############################################################################
#                         Selection Bias DAG 3: Part 1                       #
#           Simulation for the Application of Prognostic Scores              #
#                               to Selection Bias                            #
##############################################################################

dag3.dat<-function(n, gamma1, seed){
      set.seed(seed)
      Uvar<-rbinom(n, 1, 0.3)
      Evar<-rbinom(n, 1, 0.3)
      beta0<-log(.3/.7)
      beta1<-log(6)
      beta2<-log(6)
      probL<-exp(beta0+beta1*Evar+beta2*Uvar)/
            (1 + exp(beta0+beta1*Evar+beta2*Uvar))
      Lvar<-rbinom(n, 1, prob=probL)
      gamma0<-log(.3/.7)
      gamma2<-log(6)
      probY<-exp(gamma0+gamma1*Evar+gamma2*Uvar)/
            (1 + exp(gamma0+gamma1*Evar+gamma2*Uvar))
      Yvar<-rbinom(n, 1, prob=probY)
      delta0<-log(.3/.7)
      delta1<-log(6)
      probS<-exp(delta0+delta1*Lvar)/(1 + exp(delta0+delta1*Lvar))
      Svar<-rbinom(n,1, prob=probS)
      dat<-data.frame(Evar,Yvar,Svar,Lvar,Uvar)
      sel<-dat[dat$Svar==1,]
      ### Prognostic Weights
      ## based on L
      #
      # unexposed
      Lwt<- glm(Yvar ~ Lvar, data=sel[sel$Evar==0,],
                family=binomial(link="logit"))
      sel$Lwt <- predict(Lwt, newdata=sel, type="response")
      sel$Lwt[sel$Yvar==0] <- 1-sel$Lwt[sel$Yvar==0]
      L.progw<-1/(sel$Lwt)
      L.progsw <- (sel$Yvar*mean(sel$Yvar) +
                        (1-sel$Yvar)*(1-mean(sel$Yvar))) * L.progw

      # full population
      wt1 <- glm(Yvar~Lvar+Evar,data=sel,family=binomial)
```

```
tmp <- sel
tmp$Evar[tmp$Evar==1] <- 0
wt1 <- predict(wt1,newdata=tmp,type="response")
prog.l.full <- sel$Yvar*(mean(sel$Yvar)/wt1) +
      (1-sel$Yvar)*((1-mean(sel$Yvar))/(1-wt1))

# modified
wt2 <- glm(Yvar~Lvar,data=sel,family=binomial)$fitted.values
prog.l.mod <- sel$Yvar*(mean(sel$Yvar)/wt2) +
      (1-sel$Yvar)*((1-mean(sel$Yvar))/(1-wt2))

## based on U, L
# unexposed
LUwt<- glm(Yvar ~ Lvar+Uvar, data=sel[sel$Evar==0,],
           family=binomial(link="logit"))
sel$LUwt <- predict(LUwt, newdata=sel, type="response")
sel$LUwt[sel$Yvar==0] <- 1-sel$LUwt[sel$Yvar==0]
LU.progw<-1/(sel$LUwt)
LU.progsw <- (sel$Yvar*mean(sel$Yvar) +
                    (1-sel$Yvar)*(1-mean(sel$Yvar))) * LU.progw

# full population
wt3 <- glm(Yvar~Lvar+Uvar+Evar,data=sel,family=binomial)
wt3 <- predict(wt3,newdata=tmp,type="response")
prog.lu.full <- sel$Yvar*(mean(sel$Yvar)/wt3) +
      (1-sel$Yvar)*((1-mean(sel$Yvar))/(1-wt3))

# modified
wt4 <- glm(Yvar~Lvar+Uvar,data=sel,family=binomial)$fitted.values
prog.lu.mod <- sel$Yvar*(mean(sel$Yvar)/wt4) +
      (1-sel$Yvar)*((1-mean(sel$Yvar))/(1-wt4))

### IPW for selection weights
# E and L
ipw.el <- glm(Svar~Lvar+Evar, data=dat, family=binomial(link="logit"))
sel$ipw.el <- predict(ipw.el, newdata=sel, type="response")
ipw.el.w<-1/(sel$ipw.el)
ipsw.el <-mean(dat$Svar)*ipw.el.w

# E, L, and U
 ipw.elu <-glm(Svar~Lvar+Evar+Uvar,
 data=dat,family=binomial(link="logit"))
sel$ipw.elu <- predict(ipw.elu, newdata=sel, type="response")
ipw.elu.w<-1/(sel$ipw.elu)
ipsw.elu <-mean(dat$Svar)*ipw.elu.w
#
#
crude<-glm(Yvar ~ Evar, data=sel, family=binomial(link="logit")) # crude
mod1<-glm(Yvar ~ Evar, data=sel, weights=L.progsw,
          family=binomial(link="logit")) #prog L unexp
mod2<-glm(Yvar ~ Evar, data=sel, weights=prog.l.full,
          family=binomial(link="logit")) #prog L full
mod3<-glm(Yvar ~ Evar, data=sel, weights=prog.l.mod,
          family=binomial(link="logit")) #prog L mod
mod4<-glm(Yvar ~ Evar, data=sel, weights=ipsw.el,
          family=binomial(link="logit")) #ipsw L
mod5<-glm(Yvar ~ Evar + Lvar, data=sel,
          family=binomial(link="logit")) #direct adj L
mod6<-glm(Yvar ~ Evar, data=sel, weights=LU.progsw,
          family=binomial(link="logit")) #prog LU unexp
```

```
mod7<-glm(Yvar ~ Evar, data=sel, weights=prog.lu.full,
          family=binomial(link="logit")) #prog LU full
mod8<-glm(Yvar ~ Evar, data=sel, weights=prog.lu.mod,
          family=binomial(link="logit")) #prog LU mod
mod9<-glm(Yvar ~ Evar, data=sel, weights=ipsw.elu,
          family=binomial(link="logit")) #ipsw LU
mod10<-glm(Yvar ~ Evar+Lvar+Uvar, data=sel,
           family=binomial(link="logit")) #direct adj LU
mod11<-glm(Yvar ~ Evar, data=dat,
           family=binomial(link="logit")) # full population
#
## marginal OR
pbar0 <- (gamma0+gamma2*Uvar)
pbar0 <- mean(1/(1+exp(-pbar0)))
pbar1 <- (gamma0+gamma2*Uvar+gamma1)
pbar1 <- mean(1/(1+exp(-pbar1)))
marg.or <- (pbar1/(1-pbar1)) / (pbar0/(1-pbar0))
#
beta.crude<-summary(crude)$coef[2,1]
var.crude<-(summary(crude)$coef[2,2])^2
beta.mod1<-summary(mod1)$coef[2,1]
var.mod1<-(summary(mod1)$coef[2,2])^2
beta.mod2<-summary(mod2)$coef[2,1]
var.mod2<-(summary(mod2)$coef[2,2])^2
beta.mod3<-summary(mod3)$coef[2,1]
var.mod3<-(summary(mod3)$coef[2,2])^2
beta.mod4<-summary(mod4)$coef[2,1]
var.mod4<-(summary(mod4)$coef[2,2])^2
beta.mod5<-summary(mod5)$coef[2,1]
var.mod5<-(summary(mod5)$coef[2,2])^2
beta.mod6<-summary(mod6)$coef[2,1]
var.mod6<-(summary(mod6)$coef[2,2])^2
beta.mod7<-summary(mod7)$coef[2,1]
var.mod7<-(summary(mod7)$coef[2,2])^2
beta.mod8<-summary(mod8)$coef[2,1]
var.mod8<-(summary(mod8)$coef[2,2])^2
beta.mod9<-summary(mod9)$coef[2,1]
var.mod9<-(summary(mod9)$coef[2,2])^2
beta.mod10<-summary(mod10)$coef[2,1]
var.mod10<-(summary(mod10)$coef[2,2])^2
beta.mod11<-summary(mod11)$coef[2,1]
var.mod11<-(summary(mod11)$coef[2,2])^2
#
## insert calculation for % bias for marg and conditional
bias.crudecond=((beta.crude-log(3))/log(3))*100
bias.crudemarg=((beta.crude-log(marg.or))/log(marg.or))*100
bias.mod1c=((beta.mod1-log(3))/log(3))*100
bias.mod1m=((beta.mod1-log(marg.or))/log(marg.or))*100
bias.mod2c=((beta.mod2-log(3))/log(3))*100
bias.mod2m=((beta.mod2-log(marg.or))/log(marg.or))*100
bias.mod3c=((beta.mod3-log(3))/log(3))*100
bias.mod3m=((beta.mod3-log(marg.or))/log(marg.or))*100
bias.mod4c=((beta.mod4-log(3))/log(3))*100
bias.mod4m=((beta.mod4-log(marg.or))/log(marg.or))*100
bias.mod5c=((beta.mod5-log(3))/log(3))*100
bias.mod5m=((beta.mod5-log(marg.or))/log(marg.or))*100
bias.mod6c=((beta.mod6-log(3))/log(3))*100
bias.mod6m=((beta.mod6-log(marg.or))/log(marg.or))*100
bias.mod7c=((beta.mod7-log(3))/log(3))*100
bias.mod7m=((beta.mod7-log(marg.or))/log(marg.or))*100
```

```
bias.mod8c=((beta.mod8-log(3))/log(3))*100
bias.mod8m=((beta.mod8-log(marg.or))/log(marg.or))*100
bias.mod9c=((beta.mod9-log(3))/log(3))*100
bias.mod9m=((beta.mod9-log(marg.or))/log(marg.or))*100
bias.mod10c=((beta.mod10-log(3))/log(3))*100
bias.mod10m=((beta.mod10-log(marg.or))/log(marg.or))*100
bias.mod11c=((beta.mod11-log(3))/log(3))*100
bias.mod11m=((beta.mod11-log(marg.or))/log(marg.or))*100
#
## robust regression variance
library(sandwich)
robust.crude <- diag(sandwich(crude))
robust.crude <- robust.crude[2]
robust.mod1 <- diag(sandwich(mod1))
robust.mod1 <-robust.mod1[2]
robust.mod2 <- diag(sandwich(mod2))
robust.mod2 <-robust.mod2[2]
robust.mod3 <- diag(sandwich(mod3))
robust.mod3 <-robust.mod3[2]
robust.mod4 <- diag(sandwich(mod4))
robust.mod4 <-robust.mod4[2]
robust.mod5 <- diag(sandwich(mod5))
robust.mod5 <-robust.mod5[2]
robust.mod6 <- diag(sandwich(mod6))
robust.mod6 <-robust.mod6[2]
robust.mod7 <- diag(sandwich(mod7))
robust.mod7 <-robust.mod7[2]
robust.mod8 <- diag(sandwich(mod8))
robust.mod8 <-robust.mod8[2]
robust.mod9 <- diag(sandwich(mod9))
robust.mod9 <-robust.mod9[2]
robust.mod10 <- diag(sandwich(mod10))
robust.mod10 <-robust.mod10[2]
robust.mod11 <- diag(sandwich(mod11))
robust.mod11 <-robust.mod11[2]
#
## log mse calculation- used robust variance for the variance calculation
mse.crudecond=((bias.crudecond/100)^2)+robust.crude
mse.crudemarg=((bias.crudemarg/100)^2)+robust.crude
mse.mod1c=((bias.mod1c/100)^2)+robust.mod1
mse.mod1m=((bias.mod1m/100)^2)+robust.mod1
mse.mod2c=((bias.mod2c/100)^2)+robust.mod2
mse.mod2m=((bias.mod2m/100)^2)+robust.mod2
mse.mod3c=((bias.mod3c/100)^2)+robust.mod3
mse.mod3m=((bias.mod3m/100)^2)+robust.mod3
mse.mod4c=((bias.mod4c/100)^2)+robust.mod4
mse.mod4m=((bias.mod4m/100)^2)+robust.mod4
mse.mod5c=((bias.mod5c/100)^2)+robust.mod5
mse.mod5m=((bias.mod5m/100)^2)+robust.mod5
mse.mod6c=((bias.mod6c/100)^2)+robust.mod6
mse.mod6m=((bias.mod6m/100)^2)+robust.mod6
mse.mod7c=((bias.mod7c/100)^2)+robust.mod7
mse.mod7m=((bias.mod7m/100)^2)+robust.mod7
mse.mod8c=((bias.mod8c/100)^2)+robust.mod8
mse.mod8m=((bias.mod8m/100)^2)+robust.mod8
mse.mod9c=((bias.mod9c/100)^2)+robust.mod9
mse.mod9m=((bias.mod9m/100)^2)+robust.mod9
mse.mod10c=((bias.mod10c/100)^2)+robust.mod10
mse.mod10m=((bias.mod10m/100)^2)+robust.mod10
mse.mod11c=((bias.mod11c/100)^2)+robust.mod11
```

```
mse.mod11m=((bias.mod11m/100)^2)+robust.mod11

## log mse calculation- used robust variance for the variance calculation
lmse.crudecond=log(((bias.crudecond/100)^2)+robust.crude)
lmse.crudemarg=log(((bias.crudemarg/100)^2)+robust.crude)
lmse.mod1c=log(((bias.mod1c/100)^2)+robust.mod1)
lmse.mod1m=log(((bias.mod1m/100)^2)+robust.mod1)
lmse.mod2c=log(((bias.mod2c/100)^2)+robust.mod2)
lmse.mod2m=log(((bias.mod2m/100)^2)+robust.mod2)
lmse.mod3c=log(((bias.mod3c/100)^2)+robust.mod3)
lmse.mod3m=log(((bias.mod3m/100)^2)+robust.mod3)
lmse.mod4c=log(((bias.mod4c/100)^2)+robust.mod4)
lmse.mod4m=log(((bias.mod4m/100)^2)+robust.mod4)
lmse.mod5c=log(((bias.mod5c/100)^2)+robust.mod5)
lmse.mod5m=log(((bias.mod5m/100)^2)+robust.mod5)
lmse.mod6c=log(((bias.mod6c/100)^2)+robust.mod6)
lmse.mod6m=log(((bias.mod6m/100)^2)+robust.mod6)
lmse.mod7c=log(((bias.mod7c/100)^2)+robust.mod7)
lmse.mod7m=log(((bias.mod7m/100)^2)+robust.mod7)
lmse.mod8c=log(((bias.mod8c/100)^2)+robust.mod8)
lmse.mod8m=log(((bias.mod8m/100)^2)+robust.mod8)
lmse.mod9c=log(((bias.mod9c/100)^2)+robust.mod9)
lmse.mod9m=log(((bias.mod9m/100)^2)+robust.mod9)
lmse.mod10c=log(((bias.mod10c/100)^2)+robust.mod10)
lmse.mod10m=log(((bias.mod10m/100)^2)+robust.mod10)
lmse.mod11c=log(((bias.mod11c/100)^2)+robust.mod11)
lmse.mod11m=log(((bias.mod11m/100)^2)+robust.mod11)

# convert to log robust variance
lrobust.crude <- log(robust.crude)
lrobust.mod1 <-log(robust.mod1)
lrobust.mod2 <-log(robust.mod2)
lrobust.mod3 <-log(robust.mod3)
lrobust.mod4 <-log(robust.mod4)
lrobust.mod5 <-log(robust.mod5)
lrobust.mod6 <-log(robust.mod6)
lrobust.mod7 <-log(robust.mod7)
lrobust.mod8 <-log(robust.mod8)
lrobust.mod9 <-log(robust.mod9)
lrobust.mod10 <-log(robust.mod10)
lrobust.mod11 <-log(robust.mod11)

res.frame<-data.frame(beta.crude, var.crude,beta.mod1, var.mod1,
                      beta.mod2, var.mod2, beta.mod3, var.mod3,
                      beta.mod4, var.mod4, beta.mod5, var.mod5,
                      beta.mod6, var.mod6, beta.mod7, var.mod7,
                      beta.mod8, var.mod8, beta.mod9, var.mod9,
                      beta.mod10, var.mod10, beta.mod11, var.mod11,
                      marg.or, bias.crudecond, bias.crudemarg,bias.mod1c,
                      bias.mod1m, bias.mod2c, bias.mod2m, bias.mod3c,
                      bias.mod3m, bias.mod4c, bias.mod4m, bias.mod5c,
                      bias.mod5m, bias.mod6c, bias.mod6m, bias.mod7c,
                      bias.mod7m, bias.mod8c, bias.mod8m, bias.mod9c,
                      bias.mod9m, bias.mod10c, bias.mod10m, bias.mod11c,
                      bias.mod11m, robust.crude, robust.mod1,robust.mod2,
                      robust.mod3, robust.mod4, robust.mod5, robust.mod6,
                      robust.mod7, robust.mod8, robust.mod9,robust.mod10,
                      robust.mod11,mse.crudecond,mse.crudemarg,mse.mod1c,
                      mse.mod1m, mse.mod2c,mse.mod2m,mse.mod3c,mse.mod3m,
                      mse.mod4c, mse.mod4m,mse.mod5c,mse.mod5m,mse.mod6c,
```

```
                              mse.mod6m, mse.mod7c,mse.mod7m,mse.mod8c,mse.mod8m,
                              mse.mod9c, mse.mod9m, mse.mod10c, mse.mod10m,
                              mse.mod11c, mse.mod11m, lrobust.crude,lrobust.mod1,
                              lrobust.mod2, lrobust.mod3, lrobust.mod4,
                              lrobust.mod5, lrobust.mod6, lrobust.mod7,
                              lrobust.mod8, lrobust.mod9, lrobust.mod10,
                              lrobust.mod11, lmse.crudecond, lmse.crudemarg,
                              lmse.mod1c, lmse.mod1m, lmse.mod2c, lmse.mod2m,
                              lmse.mod3c, lmse.mod3m, lmse.mod4c, lmse.mod4m,
                              lmse.mod5c, lmse.mod5m, lmse.mod6c, lmse.mod6m,
                              lmse.mod7c, lmse.mod7m, lmse.mod8c, lmse.mod8m,
                              lmse.mod9c, lmse.mod9m, lmse.mod10c, lmse.mod10m,
                              lmse.mod11c, lmse.mod11m)
      res.frame

}

set.seed(754278)
#### review the issue with the seed
temp.dag3 <- matrix(NA, 1000, 1000)

output <- apply(temp.dag3, 1, function(x)
dag3.dat(1000,log(3),round(runif(1)*100000)))
dag3data<-do.call("rbind",output)

### bootstrap CIs and variance for each model
quantile(dag3data[,1],c(0.025,0.975)) # crude
crude.boot<-var(dag3data[,1])
crude.boot
quantile(dag3data[,3],c(0.025,0.975)) # mod 1
mod1.boot<-var(dag3data[,3])
mod1.boot
quantile(dag3data[,5],c(0.025,0.975)) # mod 2
mod2.boot<-var(dag3data[,5])
mod2.boot
quantile(dag3data[,7],c(0.025,0.975)) # mod 3
mod3.boot<-var(dag3data[,7])
mod3.boot
quantile(dag3data[,9],c(0.025,0.975)) # mod 4
mod4.boot<-var(dag3data[,9])
mod4.boot
quantile(dag3data[,11],c(0.025,0.975)) # mod 5
mod5.boot<-var(dag3data[,11])
mod5.boot
quantile(dag3data[,13],c(0.025,0.975)) # mod 6
mod6.boot<-var(dag3data[,13])
mod6.boot
quantile(dag3data[,15],c(0.025,0.975)) # mod 7
mod7.boot<-var(dag3data[,15])
mod7.boot
quantile(dag3data[,17],c(0.025,0.975)) # mod 8
mod8.boot<-var(dag3data[,17])
mod8.boot
quantile(dag3data[,19],c(0.025,0.975)) # mod 9
mod9.boot<-var(dag3data[,19])
mod9.boot
quantile(dag3data[,21],c(0.025,0.975)) # mod 10
mod10.boot<-var(dag3data[,21])
mod10.boot
quantile(dag3data[,23],c(0.025,0.975)) # mod 11
```

```
mod11.boot<-var(dag3data[,23])
mod11.boot

#############################################################################
#                        Selection Bias DAG 3: Part 2                       #
#           Simulation for the Application of Prognostic Scores             #
#                              to Selection Bias                            #
#############################################################################

dag3.dat<-function(n, gamma1, seed){
      set.seed(seed)
      Uvar<-rbinom(n, 1, 0.3)
      Evar<-rbinom(n, 1, 0.3)
      beta0<-log(.3/.7)
      beta1<-log(6)
      beta2<-log(6)
      probL<-exp(beta0+beta1*Evar+beta2*Uvar)/
            (1 + exp(beta0+beta1*Evar+beta2*Uvar))
      Lvar<-rbinom(n, 1, prob=probL)
      gamma0<-log(.3/.7)
      gamma2<-log(6)
      probY<-exp(gamma0+gamma1*Evar+gamma2*Uvar)/
            (1 + exp(gamma0+gamma1*Evar+gamma2*Uvar))
      Yvar<-rbinom(n, 1, prob=probY)
      delta0<-log(.3/.7)
      delta1<-log(6)
      probS<-exp(delta0+delta1*Lvar)/(1 + exp(delta0+delta1*Lvar))
      Svar<-rbinom(n,1, prob=probS)
      dat<-data.frame(Evar,Yvar,Svar,Lvar,Uvar)
      sel<-dat[dat$Svar==1,]
      ### Prognostic Weights
      ## based on L
      #
      # unexposed
      Lwt<- glm(Yvar ~ Lvar, data=sel[sel$Evar==0,],
                family=binomial(link="logit"))
      sel$Lwt <- predict(Lwt, newdata=sel, type="response")
      sel$Lwt[sel$Yvar==0] <- 1-sel$Lwt[sel$Yvar==0]
      L.progw<-1/(sel$Lwt)
      L.progsw <- (sel$Yvar*mean(sel$Yvar) +
                        (1-sel$Yvar)*(1-mean(sel$Yvar))) * L.progw

      # full population
      wt1 <- glm(Yvar~Lvar+Evar,data=sel,family=binomial)
      tmp <- sel
      tmp$Evar[tmp$Evar==1] <- 0
      wt1 <- predict(wt1,newdata=tmp,type="response")
      prog.l.full <- sel$Yvar*(mean(sel$Yvar)/wt1) +
            (1-sel$Yvar)*((1-mean(sel$Yvar))/(1-wt1))

      # modified
      wt2 <- glm(Yvar~Lvar,data=sel,family=binomial)$fitted.values
      prog.l.mod <- sel$Yvar*(mean(sel$Yvar)/wt2) +
            (1-sel$Yvar)*((1-mean(sel$Yvar))/(1-wt2))

      ## based on U, L
      # unexposed
      LUwt<- glm(Yvar ~ Lvar+Uvar, data=sel[sel$Evar==0,],
                 family=binomial(link="logit"))
      sel$LUwt <- predict(LUwt, newdata=sel, type="response")
```

```
sel$LUwt[sel$Yvar==0] <- 1-sel$LUwt[sel$Yvar==0]
LU.progw<-1/(sel$LUwt)
LU.progsw <- (sel$Yvar*mean(sel$Yvar) +
                    (1-sel$Yvar)*(1-mean(sel$Yvar))) * LU.progw

# full population
wt3 <- glm(Yvar~Lvar+Uvar+Evar,data=sel,family=binomial)
wt3 <- predict(wt3,newdata=tmp,type="response")
prog.lu.full <- sel$Yvar*(mean(sel$Yvar)/wt3) +
      (1-sel$Yvar)*((1-mean(sel$Yvar))/(1-wt3))

# modified
wt4 <- glm(Yvar~Lvar+Uvar,data=sel,family=binomial)$fitted.values
prog.lu.mod <- sel$Yvar*(mean(sel$Yvar)/wt4) +
      (1-sel$Yvar)*((1-mean(sel$Yvar))/(1-wt4))

### IPW for selection weights
# E and L
ipw.el <- glm(Svar~Lvar+Evar, data=dat, family=binomial(link="logit"))
sel$ipw.el <- predict(ipw.el, newdata=sel, type="response")
ipw.el.w<-1/(sel$ipw.el)
ipsw.el <-mean(dat$Svar)*ipw.el.w

# E, L, and U
ipw.elu <- glm(Svar~Lvar+Evar+Uvar, data=dat,
               family=binomial(link="logit"))
sel$ipw.elu <- predict(ipw.elu, newdata=sel, type="response")
ipw.elu.w<-1/(sel$ipw.elu)
ipsw.elu <-mean(dat$Svar)*ipw.elu.w
#
#
crude<-glm(Yvar ~ Evar, data=sel, family=binomial(link="logit")) # crude
mod1<-glm(Yvar ~ Evar, data=sel, weights=L.progsw,
          family=binomial(link="logit")) #prog L unexp
mod2<-glm(Yvar ~ Evar, data=sel, weights=prog.l.full,
          family=binomial(link="logit")) #prog L full
mod3<-glm(Yvar ~ Evar, data=sel, weights=prog.l.mod,
          family=binomial(link="logit")) #prog L mod
mod4<-glm(Yvar ~ Evar, data=sel, weights=ipsw.el,
          family=binomial(link="logit")) #ipsw L
mod5<-glm(Yvar ~ Evar + Lvar, data=sel,
          family=binomial(link="logit")) #direct adj L
mod6<-glm(Yvar ~ Evar, data=sel, weights=LU.progsw,
          family=binomial(link="logit")) #prog LU unexp
mod7<-glm(Yvar ~ Evar, data=sel, weights=prog.lu.full,
          family=binomial(link="logit")) #prog LU full
mod8<-glm(Yvar ~ Evar, data=sel, weights=prog.lu.mod,
          family=binomial(link="logit")) #prog LU mod
mod9<-glm(Yvar ~ Evar, data=sel, weights=ipsw.elu,
          family=binomial(link="logit")) #ipsw LU
mod10<-glm(Yvar ~ Evar+Lvar+Uvar, data=sel,
           family=binomial(link="logit")) #direct adj LU
mod11<-glm(Yvar ~ Evar, data=dat,
           family=binomial(link="logit")) # full population
#
## marginal OR
pbar0 <- (gamma0+gamma2*Uvar)
pbar0 <- mean(1/(1+exp(-pbar0)))
pbar1 <- (gamma0+gamma2*Uvar+gamma1)
pbar1 <- mean(1/(1+exp(-pbar1)))
```

```
marg.or <- (pbar1/(1-pbar1)) / (pbar0/(1-pbar0))
#
beta.crude<-summary(crude)$coef[2,1]
var.crude<-(summary(crude)$coef[2,2])^2
beta.mod1<-summary(mod1)$coef[2,1]
var.mod1<-(summary(mod1)$coef[2,2])^2
beta.mod2<-summary(mod2)$coef[2,1]
var.mod2<-(summary(mod2)$coef[2,2])^2
beta.mod3<-summary(mod3)$coef[2,1]
var.mod3<-(summary(mod3)$coef[2,2])^2
beta.mod4<-summary(mod4)$coef[2,1]
var.mod4<-(summary(mod4)$coef[2,2])^2
beta.mod5<-summary(mod5)$coef[2,1]
var.mod5<-(summary(mod5)$coef[2,2])^2
beta.mod6<-summary(mod6)$coef[2,1]
var.mod6<-(summary(mod6)$coef[2,2])^2
beta.mod7<-summary(mod7)$coef[2,1]
var.mod7<-(summary(mod7)$coef[2,2])^2
beta.mod8<-summary(mod8)$coef[2,1]
var.mod8<-(summary(mod8)$coef[2,2])^2
beta.mod9<-summary(mod9)$coef[2,1]
var.mod9<-(summary(mod9)$coef[2,2])^2
beta.mod10<-summary(mod10)$coef[2,1]
var.mod10<-(summary(mod10)$coef[2,2])^2
beta.mod11<-summary(mod11)$coef[2,1]
var.mod11<-(summary(mod11)$coef[2,2])^2
#
#
## robust regression variance
library(sandwich)
robust.crude <- diag(sandwich(crude))
robust.crude <- robust.crude[2]
robust.mod1 <- diag(sandwich(mod1))
robust.mod1 <-robust.mod1[2]
robust.mod2 <- diag(sandwich(mod2))
robust.mod2 <-robust.mod2[2]
robust.mod3 <- diag(sandwich(mod3))
robust.mod3 <-robust.mod3[2]
robust.mod4 <- diag(sandwich(mod4))
robust.mod4 <-robust.mod4[2]
robust.mod5 <- diag(sandwich(mod5))
robust.mod5 <-robust.mod5[2]
robust.mod6 <- diag(sandwich(mod6))
robust.mod6 <-robust.mod6[2]
robust.mod7 <- diag(sandwich(mod7))
robust.mod7 <-robust.mod7[2]
robust.mod8 <- diag(sandwich(mod8))
robust.mod8 <-robust.mod8[2]
robust.mod9 <- diag(sandwich(mod9))
robust.mod9 <-robust.mod9[2]
robust.mod10 <- diag(sandwich(mod10))
robust.mod10 <-robust.mod10[2]
robust.mod11 <- diag(sandwich(mod11))
robust.mod11 <-robust.mod11[2]
#
## mse calculation- used robust variance for the variance calculation
mse.crudecond=((beta.crude)^2)+robust.crude
mse.mod1c=((beta.mod1)^2)+robust.mod1
mse.mod2c=((beta.mod2)^2)+robust.mod2
mse.mod3c=((beta.mod3)^2)+robust.mod3
```

```
      mse.mod4c=((beta.mod4)^2)+robust.mod4
      mse.mod5c=((beta.mod5)^2)+robust.mod5
      mse.mod6c=((beta.mod6)^2)+robust.mod6
      mse.mod7c=((beta.mod7)^2)+robust.mod7
      mse.mod8c=((beta.mod8)^2)+robust.mod8
      mse.mod9c=((beta.mod9)^2)+robust.mod9
      mse.mod10c=((beta.mod10)^2)+robust.mod10
      mse.mod11c=((beta.mod11)^2)+robust.mod11

      ## log mse calculation- used robust variance for the variance calculation
      lmse.crudecond=log(((beta.crude)^2)+robust.crude)
      lmse.mod1c=log(((beta.mod1)^2)+robust.mod1)
      lmse.mod2c=log(((beta.mod2)^2)+robust.mod2)
      lmse.mod3c=log(((beta.mod3)^2)+robust.mod3)
      lmse.mod4c=log(((beta.mod4)^2)+robust.mod4)
      lmse.mod5c=log(((beta.mod5)^2)+robust.mod5)
      lmse.mod6c=log(((beta.mod6)^2)+robust.mod6)
      lmse.mod7c=log(((beta.mod7)^2)+robust.mod7)
      lmse.mod8c=log(((beta.mod8)^2)+robust.mod8)
      lmse.mod9c=log(((beta.mod9)^2)+robust.mod9)
      lmse.mod10c=log(((beta.mod10)^2)+robust.mod10)
      lmse.mod11c=log(((beta.mod11)^2)+robust.mod11)

      # convert to log robust variance
      lrobust.crude <- log(robust.crude)
      lrobust.mod1 <-log(robust.mod1)
      lrobust.mod2 <-log(robust.mod2)
      lrobust.mod3 <-log(robust.mod3)
      lrobust.mod4 <-log(robust.mod4)
      lrobust.mod5 <-log(robust.mod5)
      lrobust.mod6 <-log(robust.mod6)
      lrobust.mod7 <-log(robust.mod7)
      lrobust.mod8 <-log(robust.mod8)
      lrobust.mod9 <-log(robust.mod9)
      lrobust.mod10 <-log(robust.mod10)
      lrobust.mod11 <-log(robust.mod11)

      res.frame<-data.frame(beta.crude, var.crude,beta.mod1, var.mod1,
                            beta.mod2, var.mod2, beta.mod3, var.mod3,
                            beta.mod4, var.mod4, beta.mod5, var.mod5,
                            beta.mod6, var.mod6, beta.mod7, var.mod7,
                            beta.mod8, var.mod8, beta.mod9, var.mod9,
                            beta.mod10, var.mod10, beta.mod11, var.mod11,
                            marg.or, robust.crude, robust.mod1, robust.mod2,
                            robust.mod3, robust.mod4, robust.mod5, robust.mod6,
                            robust.mod7, robust.mod8, robust.mod9,robust.mod10,
                            robust.mod11, mse.crudecond, mse.mod1c, mse.mod2c,
                            mse.mod3c, mse.mod4c, mse.mod5c, mse.mod6c,
                            mse.mod7c, mse.mod8c, mse.mod9c, mse.mod10c,
                            mse.mod11c, lrobust.crude, lrobust.mod1,
                            lrobust.mod2, lrobust.mod3, lrobust.mod4,
                            lrobust.mod5, lrobust.mod6, lrobust.mod7,
                            lrobust.mod8, lrobust.mod9, lrobust.mod10,
                            lrobust.mod11, lmse.crudecond, lmse.mod1c,
                            lmse.mod2c, lmse.mod3c, lmse.mod4c, lmse.mod5c,
                            lmse.mod6c, lmse.mod7c, lmse.mod8c, lmse.mod9c,
                            lmse.mod10c, lmse.mod11c)
      res.frame

}
```

```
set.seed(754278)
#### review the issue with the seed
temp.dag3 <- matrix(NA, 1000, 1000)

output <- apply(temp.dag3, 1, function(x)
dag3.dat(1000,log(1),round(runif(1)*100000)))
dag3data<-do.call("rbind",output)

### bootstrap CIs and variance for each model
quantile(dag3data[,1],c(0.025,0.975)) # crude
crude.boot<-var(dag3data[,1])
crude.boot
quantile(dag3data[,3],c(0.025,0.975)) # mod 1
mod1.boot<-var(dag3data[,3])
mod1.boot
quantile(dag3data[,5],c(0.025,0.975)) # mod 2
mod2.boot<-var(dag3data[,5])
mod2.boot
quantile(dag3data[,7],c(0.025,0.975)) # mod 3
mod3.boot<-var(dag3data[,7])
mod3.boot
quantile(dag3data[,9],c(0.025,0.975)) # mod 4
mod4.boot<-var(dag3data[,9])
mod4.boot
quantile(dag3data[,11],c(0.025,0.975)) # mod 5
mod5.boot<-var(dag3data[,11])
mod5.boot
quantile(dag3data[,13],c(0.025,0.975)) # mod 6
mod6.boot<-var(dag3data[,13])
mod6.boot
quantile(dag3data[,15],c(0.025,0.975)) # mod 7
mod7.boot<-var(dag3data[,15])
mod7.boot
quantile(dag3data[,17],c(0.025,0.975)) # mod 8
mod8.boot<-var(dag3data[,17])
mod8.boot
quantile(dag3data[,19],c(0.025,0.975)) # mod 9
mod9.boot<-var(dag3data[,19])
mod9.boot
quantile(dag3data[,21],c(0.025,0.975)) # mod 10
mod10.boot<-var(dag3data[,21])
mod10.boot
quantile(dag3data[,23],c(0.025,0.975)) # mod 11
mod11.boot<-var(dag3data[,23])
mod11.boot
```

References

[1] Miettinen, Olli S. "Stratification by a multivariate confounder score." *American Journal of Epidemiology* 104.6 (1976): 609-620.
[1] Hansen, Ben B. "The prognostic analogue of the propensity score." *Biometrika*95.2 (2008): 481-488.
[1] Arbogast, Patrick G., and Wayne A. Ray. "Performance of disease risk scores, propensity scores, and traditional multivariable outcome regression in the presence of multiple confounders." *American Journal of Epidemiology* 174.5 (2011): 613-620.
[1] Arbogast, Patrick G., and Wayne A. Ray. "Use of disease risk scores in pharmacoepidemiologic studies." *Statistical methods in medical research* 18.1 (2009): 67-80.
[1] Stuart, Elizabeth A., Brian K. Lee, and Finbarr P. Leacy. "Prognostic score–based balance measures can be a useful diagnostic for propensity score methods in comparative effectiveness research." *Journal of clinical epidemiology* 66.8 (2013): S84-S90.
[1] Tadrous, Mina, et al. "Disease risk score as a confounder summary method: systematic review and recommendations." *Pharmacoepidemiology and drug safety* 22.2 (2013): 122-129.
[1] Pike, M. C., J. Anderson, and N. Day. "Some insights into Miettinen's multivariate confounder score approach to case-control study analysis."*Epidemiology and community health* 33.1 (1979): 104-106.
[1] Francis Cook, E., and Lee Goldman. "Performance of tests of significance based on stratification by a multivariate confounder score or by a propensity score." *Journal of clinical epidemiology* 42.4 (1989): 317-324.
[1] Leacy, Finbarr P., and Elizabeth A. Stuart. "On the joint use of propensity and prognostic scores in estimation of the average treatment effect on the treated: a simulation study." *Statistics in medicine* (2013).
[1] Stuart, Elizabeth A. "Matching methods for causal inference: A review and a look forward." *Statistical science: a review journal of the Institute of Mathematical Statistics* 25.1 (2010): 1.
[1] Hernán, Miguel A., Sonia Hernandez-Diaz, and James M. Robins. "A structural approach to selection bias." *Epidemiology* 15.5 (2004): 615-625.
[1] Greenland, Sander. "Quantifying biases in causal models: classical confounding vs collider-stratification bias." *Epidemiology* 14.3 (2003): 300-306.
[1] Howe, Chanelle J., et al. "Limitation of inverse probability-of-censoring weights in estimating survival in the presence of strong selection bias." *American journal of epidemiology* 173.5 (2011): 569-577.
[1] Berkson, Joseph. "Limitations of the application of fourfold table analysis to hospital data." *Biometrics Bulletin* (1946): 47-53.
[1] Gordis, Leon. Epidemiology. Philadelphia: Elsevier/Saunders, 2009. 206-208.Print.
[1] R Core Team (2012). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.
[1] Daniel E. Ho, Kosuke Imai, Gary King, Elizabeth A. Stuart (2011). MatchIt: Nonparametric Preprocessing for Parametric Causal Inference. Journal of Statistical Software, Vol. 42, No. 8, pp. 1-28. URL http://www.jstatsoft.org/v42/i08/
[1] Austin, Peter C. "The performance of different propensity score methods for estimating marginal odds ratios." *Statistics in medicine* 26.16 (2007): 3078-3094.
[1] Achim Zeileis (2004). Econometric Computing with HC and HAC Covariance Matrix Estimators. Journal of Statistical Software 11(10), 1-17. URL http://www.jstatsoft.org/v11/i10/.
[1] Achim Zeileis (2006). Object-oriented Computation of Sandwich Estimators. Journal of Statistical Software 16(9), 1-16. URL http://www.jstatsoft.org/v16/i09/.
[1] Rosenbaum, Paul R., and Donald B. Rubin. "The central role of the propensity score in observational studies for causal effects." *Biometrika* 70.1 (1983): 41-55.
[1] Stuart, Elizabeth A. "Matching methods for causal inference: A review and a look forward." *Statistical science: a review journal of the Institute of Mathematical Statistics* 25.1 (2010): 1.

Keri Lee Calkins

DOB: 07/01/1987
103 E. Mount Royal Ave Apt 301
Baltimore, MD 21202
Cell: (713) 562-2902
e-mail: kcalkins@jhsph.edu

EDUCATION AND TRAINING

Expected May 2014	**Master of Science (ScM)**, Department of Epidemiology, Concentration in Methodology, Johns Hopkins Bloomberg School of Public Health (JHSPH), Baltimore, MD Adviser: Dr. Bryan Lau
May 2009	**Bachelor of Arts (BA)**, Public Health Studies, Spanish Language and Culture, Johns Hopkins University (JHU), Baltimore, MD Dean's List: Fall 2005, Fall 2006, Fall 2008

PROFESSIONAL EXPERIENCE

June 2013-Present Public	**Research Assistant**, Johns Hopkins Bloomberg School of Health, Baltimore, MD -Working with Dr. Bryan Lau on research for the Women's Interagency HIV Study. Conducts simulations and statistical analysis in R focused on selection bias.
July 2010-August 2012	**Research Data Specialist,** Dana Farber Cancer Institute, Boston, MA - Tracked outcomes of Hematopoietic Stem Cell Transplants (HSCT) in various clinical research databases. Developed and implemented recurring data quality assurance projects for three HSCT research databases using Microsoft Excel and Access programs for 3,000,000+ data points. Trained data management staff on common data reporting errors and data management best practices. Prepared presentations for HSCT physicians on survival, relapse, and transplant complications data.
August 2009-April 2011	**Field Researcher**, Client Education Initiative- Sinapi Aba Trust, Kumasi, Ghana - Designed an entrepreneurship and public health curriculum for 90,000 low-literacy microfinance clients. Coordinated research activities and data collection between Ghana and US research teams.
Jan 2007-May 2009	**Student Advocate**, Lift (formerly National Student Partnerships), Baltimore, MD

	-Worked one-on-one with low-income clients to facilitate goal planning, resource access, and career development.
Sept 2007-May 2008	**Research Assistant**, Johns Hopkins Bloomberg School of Public Health, Baltimore, MD -Worked in Dr. Valeria Culotta's lab preparing media and sanitizing equipment for research activities.
Summer 2007	**Hospital and Community Intern**, Shyira Hospital, Shyira, Rwanda -Developed a medical record database and performed an analysis of a pediatric HIV/AIDS intervention program. Taught MS Excel to hospital physicians and observed rounds and surgeries. Taught entrepreneurship and pre-K classes for local students.

SKILLS

Foreign Languages: Spanish – Proficient reading/writing, Conversational speaking
Computing skills: R, STATA, SAS, Microsoft Excel, Microsoft Access

TEACHING

March 2014- May 2014	**Teaching Assistant,** Johns Hopkins Bloomberg School of Public Health, Baltimore, MD -Courses: Methodological Challenges in Epidemiologic Research (Dr. Thomas Glass, Dr. Alison Abraham, Dr. Bryan Lau)
Aug 2013- March 2014	**Lead Teaching Assistant,** Johns Hopkins Bloomberg School of Public Health, Baltimore, MD -Courses: Epidemiology and Natural History of Human Viral Infections, Epidemiology and Public Health Impact of HIV/AIDS (Dr. Homayoon Farzadegan)

HONORS AND AWARDS

2005-2009	AT&T Foundation Scholarship

SUBMITTED PUBLICATIONS

(Submitted) Eric B Schneider, PhD; **Keri L Calkins, BA**; Matthew J Weiss, MD; Joseph M Herman, MD; Christopher L Wolfgang, MD, PhD; Martin A Makary, MD, MPH; Nita Ahuja, MD; Adil H Haider, MD, MPH; Timothy Pawlik, MD, MPH, PHD. Race-Based Differences in Length-of-Stay among Patients Undergoing Pancreaticoduodenectomy. *Surgery*. 2014.

ABSTRACTS

Schneider E. B., **Calkins K. L.,** Weiss M. L., Wolfgang C. L., Makary M. A., Ahuja N., Haider A. H., Pawlik T. M. Black and Hispanic Pancreaticoduodenectomy Patients Are

Treated by Lower Volume Providers and Have Longer Hospital Stays Compared with White Patients [Abstract]. American Surgical Congress 2014; 4-6 Feb 2014; San Diego, CA. Abstract Control Number: ASC20140363

CONFERENCE PRESENTATIONS

Calkins K., Fackler L., Gammerman S. Ways to Obtain Data from Referring Physicians. CIMBTR/NMDP Clinical Research Professionals Data Management Conference 2011 [Conference]. 3 Nov 2011; Minneapolis, MN.

COMMUNITY SERVICE/LEADERSHIP

Sep 2013-May 2014 — **Community Service Co-chair**, Epidemiology Student Organization, Johns Hopkins Bloomberg School of Public Health, Baltimore, MD
- Network with local NGOs to find service opportunities for epidemiology students. Helps to organize a teaching and mentorship project for high school students to introduce them to careers in public health.

Nov 2012-May 2014 — **Chapter Council Adviser**, Kappa Kappa Gamma, Johns Hopkins University, Baltimore, MD
- Mentors 18 undergraduate officers of Kappa Kappa Gamma at JHU. Attends weekly meetings to provide feedback and act as a mediator between officer disputes. Developed and implemented an alcohol education program based on harm reduction principles attended by 100 members of the sorority.

Aug 2012-May 2014 — **Member/Event Coordinator**, Baltimore Student Harm Reduction Coalition, Baltimore, MD
- Organized 2 annual symposiums and annual lecture series including budget preparation, fundraising, advertisement, and scheduling. Participates and leads various volunteer and advocacy projects in Baltimore focused on harm reduction principles including street outreach to sex workers and organizing HIV advocacy town halls.